

Capturing Negation in Question Answering Systems

Maria Gavriel

Supervisors: Bonnie Webber and Johan Bos



Master of Science

in

Speech and Language Processing

Theoretical and Applied Linguistics

School of Philosophy, Psychology and Language Sciences

University of Edinburgh

2005

Abstract

This masters thesis will present my research on the world of Negation, as well as my attempt to apply the Negation Algorithm on a new Question Answering system which is able to accept negative input in natural language. The aim of this project is to focus on the uses of negation in natural language, and on the importance of including negative constructions in Information Retrieval processes, which for the moment treat negation as a nonexisting phenomenon in natural language.

The new restricted-domain question answering system is called NotFilms, and accepts subject and object questions regarding Movies of 2005. NotFilms reads the input in natural language, produces its semantic representation, applies the IR algorithm on the semantic reading, and provides the user with the exact answer. It allows the existence of the negative particle 'not' in the input, and as long as the input can be semantically represented by the linguistic processes of the system, it answers both affirmative and negative questions with the same efficiency. Results have shown that the linguistic and IR processes of the system can give relevant answers for 75% of the users' questions.

Acknowledgements

Many thanks to my supervisors Bonnie and Johan for their time, my sister for her proofreading services, my family for their patience and support, and Demetris for being there for me at all times.

I also thank Silke, the dear friend who was brave enough to pose questions to my system first.

A special *Thank you* to the numerous friends and family who would come to drag me out of the house every now and then.

I keep the last *thank you* to Father Raphael, of the Edinburgh Orthodox church, for helping me overcome my fears and worries in difficult emotional times, by reminding me the small and insignificant everyday moments, which fill our lives with happiness.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text. This work has not been submitted for any other degree or professional qualification except as specified.

(Maria Gavriel)

To the victims of Helios Airways flight HCY522,
that crashed in Greece, traveling from Cyprus to Athens and then to Prague,
on Sunday, August 14, 2005.

Table of Contents

1	Introduction	1
2	What do we mean with Negation?	5
2.1	Theory of Negation: <i>What part of 'NO' do you not understand?</i> . .	5
2.2	Types of Negation	8
2.3	Conclusion	10
3	Information Retrieval and Question Answering	13
3.1	Information Retrieval: An Overview	13
3.2	The mistreatment of semantically significant words	15
3.3	Negation in Search Engines	17
3.4	Question Answering Systems: <i>The State of the Art</i>	19
3.5	Conclusion	21
4	System Requirements	23
4.1	Question Answering and Databases	23

4.2	Functional Requirements	24
4.3	CURT Modules Reuse	24
4.4	Knowledge Base	26
4.4.1	Information Elicitation	27
4.5	Prolog Facts Construction	29
4.6	Domain-Specific Negative Constructions	31
4.6.1	Small Scale Exploratory Study	31
4.6.2	Focus on Specific Constructions	33
4.7	Conclusion	34
5	Implementation	35
5.1	System Architecture Components	35
5.2	Input Component	37
5.2.1	Move <i>Not</i> to the appropriate position	37
5.2.2	Disambiguate the pronoun <i>Who</i>	38
5.3	Processing Component: <i>Linguistic Analysis</i>	38
5.3.1	Define the POS of individual words	39
5.3.2	Parse the Question	47
5.3.3	Produce the Semantic Reading of the Question	49
5.4	Processing Component: <i>The Negation Algorithm and IR Processes</i>	50
5.4.1	The Negation Algorithm	52

5.4.2	Retrieving Information with FIND/3	54
5.5	Output Component	62
5.6	Conclusion	64
6	Examples	65
6.1	Subject Questions	65
6.2	Object Questions	68
6.3	Questions with Coordinations	69
6.4	Conclusion	71
7	System Evaluation	73
7.1	Rephrasing the Collected Questions	74
7.2	Users' Evaluation	77
7.3	Conclusion	78
8	Conclusions	79
8.1	NotFilms Capabilities	79
8.2	Next Steps - Future Work	80
8.3	Conclusion	81
A	List of Questions	83
B	The entries which were presented to evaluators	87

Chapter 1

Introduction

Socrates: Tell us further, Melitus, in the name of Jupiter, whether is it better to dwell with good or bad citizens? Answer, my friend; for I ask You **nothing** difficult. Do **not** the bad work some evil to those that are continually near them, but the good some good?

Melitus: Certainly.

Socrates: Is there any one that wishes to be injured rather than benefited by his associates? Answer, good man; for the law requires you to answer. Is there any one who wishes to be injured?

Melitus: No, surely.

Socrates: Come, then, whether do you accuse me here, as one that corrupts the youth, and makes them more depraved, designedly or **undesignedly**?

Melitus: Designedly, I say.

Socrates: What, then, Melitus, are you at your time of life so much wiser than I at my time of life, as to know that the evil are always working some evil to those that are most near to them, and the good some good; but I have arrived at such a pitch of ignorance as **not** to know that if I make any one of my associates **depraved**, I shall be in danger of receiving some evil from him; and yet I designedly bring about this so great evil, as you say? In this I **can not** believe you, Melitus, **nor** do I think would **any other** man in the world. But either I **do not** corrupt the youth, or, if I do corrupt them, I do it **undesignedly**: so that in both cases you speak **falsely**. But if I corrupt them **undesignedly**, for such **involuntary** offenses it is **not usual** to accuse one here, but to take one apart, and teach and admonish one. For it is evident that if I am taught, I shall **cease** doing what I do **undesignedly**. But you shunned me, and **were not** willing to associate

*with and instruct me; but you accuse me here, where it is usual to accuse those who need punishment, and **not** instruction.*

[Excerpt from Plato's 'The Apology of Socrates]

My quest of finding some excerpt that would concisely demonstrate different constructions of negation has ended up quite discouraging. Random pages from novels, science fiction, detective or thriller books, youth magazines, scientific journals or news articles, rarely have more than one negative structure or particle grouped together. Instead, negative language was more likely to be found in reviews (where the reviewer criticizes that piece of work), columns of political analysts who again disapprove a certain politicians or governments actions, or even philosophical texts.

Indeed, the excerpt I chose to quote as the set off point of my dissertation, comes from a translation of 'The Apology of Socrates, written by Plato ¹. The excerpt is the eleventh dialogue between Socrates and his main accuser, Melitus.

The goal of my project is to investigate the theory of negation and how this is incorporated in every-day language. Working in the fields of computational linguistics, my attempt will be to bring together theory and practice; that is to realise instances of negation in language and try to develop techniques that would allow negative questions in the form of natural language to be accepted, analysed, as well as correctly answered by an existing Question Answering system.

Information Retrieval [IR] and Question Answering [QA] systems do not have much to illustrate about negation usages. Although QA systems are developing fast through processing more and more question types from factoid questions to multimodal, more philosophical-oriented questions, there has hardly

¹The translation of the Socratic dialogues can be accessed online at <http://www.gutenberg.org/dirs/1/3/7/2/13726/13726-8.txt>

been any attempt to include techniques which would focus on the processing of negated questions. Free trials of small questions like 'Which directors did not direct thrillers?' on Google², Yahoo³, Ask Jeeves⁴, and the likes, demonstrate this claim. What is at least needed for such an attempt, is some shallow linguistic parsing of the sentence, to indicate what is really negated, and thus not wanted by the user to be shown.

The state of affairs of not making specialised attempts to deal with negative input has resulted in the lack of references about applications of negation in question answering systems. I hope that this piece of work will be a reference point for future researchers on the area of negation. Throughout this work, a question answering system that is able to read, parse and give the right answer to a number of negative questions will be introduced. The system, which is called 'NotFilms', accepts domain-specific questions regarding information about movies of 2005.

The following chapter will introduce the reader to theories of negation by different linguists and philosophers. Chapter 3 will present the capabilities of current Question Answering systems, and where negation could fit in. In Chapter 4, I am describing the preparations before putting together a new question answering system which accepts subject and object questions: from how I chose to use ready modules from an existing system, to how I collected real domain-specific questions, and to the creation of the knowledge base. Chapter 5 presents the architecture of the created system, with a focus on how the negation algorithm works. Chapter 6 includes examples of questions which NotFilms can efficiently answer, based on the input's semantic interpretation and the stored information in the databases. The evaluation process of the negation system is found in Chapter 7. It shows that although most of the structures of the questions cannot be captured from the first time, and thus need rephrasing by the users, still, after some rephrasing, the system can cor-

² www.google.com

³ www.yahoo.com

⁴ www.ask.com

rectly retrieve the answer for 75% of the users' questions. Chapter 8 is reserved for considerations and suggestions for future work, and is followed by some concluding remarks.

Chapter 2

What do we mean with Negation?

Negation has long been studied, or at least approached by the fields of logic, linguistics, psychology, philosophy of mind and cognitive science. It has been seen in comparison to (or contrasted with) the affirmative, yet it has hardly been analyzed as an issue of language by itself. The challenge of negation begins with its own definition, or rather with the lack of a 'proper definition.

2.1 Theory of Negation: *What part of 'NO' do you not understand?*

Attempting to give a definition of negation which does *not* include a negative marker, one realises that it is hard to achieve. In the study of language, although the affirmative has been approached in a straight forward way, negation is extremely difficult to define or describe (Downing, 2000, p.23).

The consensus of opinions is that, structurally, negation is far more complex than the corresponding affirmative, and that this complexity can be mirrored by the different angles which negation has been examined from (Downing,

2000, p.36). In addition, negative statements are found in more complex situations. They are mostly apparent when the speaker denies what would be the obvious, an expectation (Wason, 1965): If a tourist describing her day out in Edinburgh said: 'I have not visited the Haymarket palace.' this would proclaim an expectation of either the speaker or the hearer, that the speaker would have visited the palace. By using negation, this expectation is denied.

Experience is believed to be coded in the mind in an affirmative way (Clark and Clark, 1977). This observation has been made from narrative studies, observing the attitudes of people telling stories about events that took place sometime in the past. The same observation was made from descriptive studies, on people who were describing places or buildings. Almost always, the observants were using positive language to describe a place or tell a story, and would under rare occasions use negative language, only to point out something that happened or appeared differently than what would have been expected under normal circumstances (Wason, 1965; Clark and Clark, 1977).

In the literature, there is a view called affirmation-negation symmetry (Jordan, 1998). According to the symmetry approach, affirmative and negative processes walk side by side. The negative requires a previous existence of the positive, yet at the same time, the positive requires the realisation of its negative in order to stand out. Nevertheless, an increased number of researchers support that negative constructions are second degree operations which are built after pre-existing positive structures (Clark, 1976), and thus are quite more complex for the brain to process, revealing some asymmetry between the two operations. Experimental evidence (Volterra and Antinucci, 1979) in (Downing, 2000) has suggested that young children use negative constructions in situations that presuppose a corresponding positive statement. Even Aristotle had views to share on this issue: That there are two reasons why the affirmative is prior to the negative: In speech, affirmation is simpler and more readily said, for the obvious reason that negation needs a negative marker or particle to be added to the affirmative. Second, in the mind, an affirmative expres-

sion requires composition of thoughts, whereas a negative expression requires a division between the negative expression and the counter affirmative, and distancing one's self from the latter. (Oesterle, 1962).

According to the contemporary scholar, Domenico Pacitti, negation aims to deny the affirmative proposition it contains. In the following example,

Example 1: *Tomorrow's football match will **not** take place at five o'clock.*

negation goes further than mere making a proposition. It rather makes an act of judgement over the included affirmation. It contains the affirmation, and it rejects it. Even if the affirmative sentence might also make some sort of judgement, still, negation would be judging on the affirmative judgement, making a second order judgement, thus being at a higher level of cognitive processes (Pacitti, 1991). The scholar supports that this higher level of negative judgement carries by definition more dynamism compared to a positive one, which is generally seen as more stationary.

Pacitti (1991) also deals with the fact that judgement has to do with differentiating the one thing from the rest, as well as questioning relations between things. In the above example (example 1), the meaning of the sentence is actually ambiguous. The next two sentences will disambiguate the meanings:

- (a) *Tomorrow's match is **not** at 5pm, but at 7 o'clock tomorrow.*
- (b) *Tomorrow's match will **not** take place at 5pm, because there is simply no match tomorrow.*

Consequently, what would be denied in the first case would be a mistaken judgement over the existing situation, but in the second case, the whole pre-supposition of the affirmative judgement, regarding tomorrow's match will be rejected (Pacitti, 1991). The dynamics of negation outperform the fixity of affirmation.

Due to the structural complexity of negation, children are believed to acquire

and use negative constructions in language long after they start using affirmative language. Nevertheless, it is known that even before verbal expressions of negation, children do have cognitive awareness of negation, expressed by non-linguistic means that might denote refusal, non-existence, etc. (Clark and Clark, 1977).

What has been presented so far is a grasp of the philosophy behind negation in general, approached by scientists of different fields. Nevertheless, every theory almost always evolves after actual observations on the issue in question. Negation is part of people's everyday language. They use it broadly in casual or formal conversations, mirroring their cognitive abilities of critical thinking. The content of the following section is about the realisation of negation in language. Classifying negation in different types is the starting point for analysts of any science to enquire into the nature of negation.

2.2 Types of Negation

In English, there are different ways of expressing negation depending on the situation, the emphasis of the sentence, the style of the speaker, etc. Nevertheless, here as well, classifying the several negative expressions into different types has neither been a piece of cake for the researchers, nor has it been straightforward, so for them to come to a broad (at least) agreement. There has been an extended interest in the issue (Klima, 1964; Clark and Clark, 1977; Tottie, 1991; Givon, 1993), and many more, with different suggestions for such a classification. I will not go into detail as to what kinds of different approaches there are to this issue. Instead, I have chosen one approach, which I found indeed uncomplicated and easy to grasp, and it is the one mentioned in (Givon, 1993). His three types of negation are explained below and are illustrated with example sentences:

Syntactic: What Givon (1993) calls syntactic negation includes words which

have a syntactically -and not morphologically- realised negation. The negative markers of this type stands words by themselves. These words can be determiners (*no, any*), pronouns (*none, nothing, nobody*), adverbs (*never, rarely, not*), etc.:

- a. *Who is **not** going to the movies this evening?* (preposition)
- b. *Are there **no** summer visitors this year?* (determiner)
- c. *That poor child has **nobody** left in the world.* (pronoun)

Morphological: This type of negation captures every English word which has a negative meaning due to an affix attachment. Thus the property of negation is carried with bound morphemes, attached to different parts of speech. Words with suffixes like *-less*, or prefixes like *un-*, *dis-*, *in-*, *il-*, *im-*, and so on, (e.g. *speechless, unmarried, dislike, inappropriate, illegal, impossible, etc.*) fall in this type of negation:

- a. *The father made sure that his daughter would depict his **disapproval** about her marrying such a person.* (prefix)
- b. *They called to book the single room for two nights, but did you tell them that it is still **unavailable**?* (prefix)
- c. *I am sorry to say it, but I find what you do completely **pointless**.* (suffix)

Inherent: This is the third and less straight forward type of Givon's negation. There is no linguistic 'rule' that would classify certain words in this category, simply because the property of negation is only detectable from the connotations of the word itself. Words of this type are *deny, fail to*, etc. along with their derivatives, of course. The negation is incorporated in the semantics of the actual words, and they do not need additional negative markers to denote the negative meaning:

- a. *She was devastated to hear that they **had rejected** her.*
- b. *Was it really on the newspapers that last night's concert was a complete*

failure?

- c. *The fire fighter was shattered to realise that he **had failed to** save the last victim from the fire.*

As can be seen from the above examples, the negated words of the first two types of Givon (1993) can easily be distinguished, either in syntactic or in morphological terms. As for the difference in meaning between the two ways, Reyes et al. (1999) assert that what they call ‘predicate negation’ (‘not honest’) (Givon’s first type) differs in meaning from ‘predicate term negation’ (‘dishonest’) (Givon’s second type), in terms of assertion strength. The ‘predicate term negation’ is considered stronger than the ‘predicate negation’ (Reyes et al., 1999, p.242).

Regarding the third type of negation, there is no regularity in the words that belong to it. For this reason, as Tottie (1991) has commented, trying to capture inherent negative words from a corpus is indeed practically impossible (Tottie, 1991). But how helpful is that from the perspective of computational linguistics? The lack of regularity in the appearance of inherently negative words is a major drawback for the development of linguistically intelligent systems that accept users’ input in natural language.

2.3 Conclusion

Without exception, all research on negation, be it philosophically, psychologically, or linguistically driven, has come to the conclusion that negation is very complex and hard to define. The fact that there are many conflicting theories regarding the cognitive processes behind negation in language, only reinforces the belief that issues behind the processes of negation have remained unresolved. The statement of Jamie Tappenden in the late nineties at the opening of his article when talking about denials and negation in English, is indicative of the growing confusion on the matter:

The ultimate objective is less to solve certain problems than to create others, by illustrating how the issues that form the topic of this paper are more intricate than previously realised, and that they are related in delicate and somewhat surprising ways.

(Tappenden, 1999)

In this chapter, we have visited different areas of the theory and realisation of negation. The attempt was to abridge to the reader the blurry and complicated phenomenon of negation, especially when it comes to depict negation in examples. In the next chapter, we will visit the area of Question Answering systems, see how far they have gone as regards the analysis of language, and yet how far they have not gone, in terms of dealing with negation in language.

Chapter 3

Information Retrieval and Question Answering

This chapter will begin with an overview of Information Retrieval Systems. It will then discuss the negligence of small common words by IR systems, and continue with the way systems incorporate negation. It will finish with a discussion about the state of the art of Question Answering Systems.

3.1 Information Retrieval: An Overview

Information Retrieval relates to the task of retrieving documents in response to a user's need of information (Jurafsky and Martin, 2000). In IR, a document refers to any different kind of text: from a paragraph, to a newspaper article, a web page, a whole website, or even a book. As an extend, a collection refers to a set of what each retrieval system wants to define as documents, choosing any of the pre-mentioned forms, or more.

Each Information Retrieval System is built with different specifications, which have to do with:

1. the kinds of documents the system works on;
2. and thus the appropriate document collection;
3. the form of queries, meaning the kind of input from the user, in order to find out the information s/he is looking for;
4. choosing the documents which are relevant to the user's query;
5. making a sub-choice of the documents which include the answer that interests the user (called the 'answer extraction' step);
6. and finally, returning the results to the user in the appropriate format. This needs to be a choice on the order of the documents, if there is more than one answer, and the actual presentation of the results (a list of links that contain the answer, a graph, etc.).

A query can be a single word, a list of words, an undivided phrase like 'Republic of Bolivia'; it can include Boolean operators (AND, OR, etc.), non Boolean operators (SITE referring to a website, etc.) (Russell and Norvig, 2003, p.841). A query can also be realized as a sentence in natural language, in the form of a question posed by one human to another. For example: 'What information do you have available about "The Republic of Bolivia"?' Besides, the acceptance of queries in natural language has been the focus of Recent Information Retrieval systems, like Question Answering systems, as we will see in the following section.

However, natural language input (also known as free input) is indeed hard to process. The difficulty lies on the multiple layers of linguistic analysis the system would need to go through:

- Syntactic analysis (i.e. word order),
- Morphological analysis (e.g. singular vs. plural form),
- Semantic analysis (e.g. definite vs. indefinite article),
- Pragmatic analysis (e.g. dealing with the problem of Pronoun Resolution,

which has to do with specifying what a pronoun refers to):

*I met the Browns and the Smiths today. Did you know that **they** bought a new house in Aberdeen?*

That is an example of a pragmatic ambiguity regarding the pronoun 'they': Whether the pronoun refers to the Browns, the Smiths, or both, we cannot suggest any definite answer. Consequently, trying to implement an IR system which would be able to deal with such ambiguous Natural Language input is way too complicated.

A starting point for the IR systems which accept natural language input, is to choose between 'key words' and non key words from the user's input, and only move on to the retrieval phase having shrunk the user's request. The following section will discuss the pros and cons of this method.

3.2 The mistreatment of semantically significant words

In order to facilitate the look up process of Information Retrieval Systems, a set of common words with 'little information content', as (Russell and Norvig, 2003, p.848) describe them, are removed from the input. Such words, which are generally referred to as 'stop words', are usually derived from closed class linguistic categories, such as determiners, prepositions, etc. They are high frequency words which make the process of term selection much slower. Therefore, since determiners and prepositions are not thought to carry informative meanings, and are thus not influencing the process of retrieving texts, they are omitted. In order to make the retrieval process quicker, even adverbs are sometimes included in the list of stop words, like for example 'where', 'when', etc.¹

Nevertheless, even if determiners such as 'a' and 'the' are disregarded by IR

¹About Google <http://www.google.com/help/basics.html>

systems, like for example the internet search engines, they carry distinct semantic interpretation, which is consequently lost with their removal of the search. Consider this set of sentences:

1. *I've seen the thriller.*
2. *I've seen a thriller.*

The difference between the first and the second sentence above, is that the first one presupposes the existence of just one thriller, whereas the second one presupposes the existence of at least one thriller (but there could be more as well). The connotations of the two determiners are so distinctive, that each one has a different representation in First Order Logic (\exists for the determiner 'a' and \forall for the determiner 'the'). Nevertheless, the IR system that classifies the above determiners as stop-words, will fail to realise any difference in meaning whatsoever.

However, the banning of semantically significant closed-class words from the IR input is not the real issue here. What really is problematic, is the processing of the input by the system, in order to extract the relevant documents for the user's needs. Once the user's input is there, the system first removes any stop words, then reads the remaining words, and then goes in the documents to retrieve the ones which contain key words from the input. No linguistic analysis is actually done, and that is where the problem lies. Consider the following two sentences:

1. *Which actresses have played in T.V. series?*
2. *Which actresses have **not** played in T.V. series?*

Here, the addition of the negative word 'not' changes the meaning of the sentence. Fortunately the word 'not' does not belong to the stop list of IR systems. Unfortunately, though, the word by itself does not serve any linguistic purpose for the IR process; it is just a word, carrying even less weight than the key words 'actresses' or 'T.V. series'. The answer for the two questions is not

the same, and definitely, a person uttering the first sentence, would not want to hear the answer of the second question, and vice versa. However, this problem has remained unsolved. The retrieved documents for each question will not differ from one another.

The next section will discuss the extend to which search engines - the most widely known Information Retrieval systems - incorporate negation in their retrieval processes.

3.3 Negation in Search Engines

Some of the most known examples of Information Retrieval systems are online search engines, like Google, Yahoo, MSN Search², Teoma³, etc. All of these systems make use of stop words in order to eliminate the spam of their search, thus making the answer extraction more efficient and less time-consuming. They then read the remaining words from the user's input, and try to match them to documents found in the World Wide Web.

Some feature for negation is indeed used, when the user wants to exclude unwanted words from the retrieved documents. The use of 'NOT' in capital letters, or the minus '-' sign before a word in the input language, indicates that what follows the negative marker should not be found anywhere in the retrieved text. This option of search is used in Google, Yahoo⁴, MSN Search⁵, etc. The negative feature is rather used in the mathematical way of minus, meaning 'subtract X from Y when you see Y -X'.

This option is useful when a user makes a search on a word which has more than one meanings. For instance, a user might want to find information about

²search.msn.com

³www.teoma.com

⁴About Yahoo <http://help.yahoo.com/help/us/ysearch/basics/basics-04.html>

⁵About MSN Search http://search.msn.com/docs/help.aspx?t=SEARCH_REF_AdvSrch-Operators.htm&FORM=HTHP#2

basses. What s/he is interested in, is the fish 'bass' and not the musical instrument 'bass'. Just googleing 'bass', there will be 21 million retrieved documents, most of which will be irrelevant to the user's search. If the user writes 'bass -music', the number of retrieved documents falls drastically to half.

Under circumstances of ambiguous meanings, the option of using the minus sign might indeed be very helpful for the user. However, when more processing is required, the search engines fail to treat negation in a satisfactory way. There is no way for a user to find the answer of 'directors that did not direct thrillers' from a search engine. There is the choice of writing the request in inverted commas. This will turn the input into a pattern, and try to retrieve documents which include all, or at least most of the given words in the same word order. However, for the same request, typing "directors that did not direct thrillers" is a very big pattern to match to anything out there, and brings zero results. The minus sign does not serve any purpose at all in this situation, because one cannot really write 'directors -thrillers' or 'direct -thrillers' and maintain the meaning of the user's request. The more abstract the input is, the more documents will be retrieved. And having the user go through the whole list of 100-200 retrieved documents in order to realise that the answer is still not there, means that the system has failed to help the user in any way.

This section has shown the use of the negation option by known search engines. The aim was to point out that the interpretation of this option by the IR systems has nothing to do with the linguistic interpretation of negation. Clearly, this mathematical way of treating negation is not a satisfactory way of capturing negation in natural language as a whole, and thus negation -in the linguistic sense- is absent from these Information Retrieval systems. In the next section, there will be a demonstration of the state of current Question Answering systems.

3.4 Question Answering Systems: *The State of the Art*

Questions progress from simple factoid questions to how to why then to what-if questions, whereas answers start out as simple facts but move to scripted or templated answers and then progress further to include multi-modal answers.

(Maybury, 2002)

Readers who read this quote, describing a Question Answering Roadmap of the recent years, might stand in awe in front of the rapid development of question answering systems. This roadmap is a plan of how systems are likely to evolve until 2006, and was written in 2002. Indeed, the claim regarding the focus of current and close-future research sounds very optimistic. Straight-forward factoid questions, like:

'Which country has the largest part of the Amazon rain forest?'

would require one and only simple answer: *Brazil*.⁶

Nevertheless, moving on to questions that would require more complex reasoning in order for the system to retrieve the answer, like *'How to...?' or 'What if...?'* questions, does not necessarily mean that the area of 'simple' factoid questions has been covered thoroughly. A similar question to the one above, could be:

'Which country does not have a part of the Amazon rain forest?'

The semantics of the sentence changes because of the appearance of 'not'. The reason why such a negative sentence needs more in depth analysis by the Question Answering system, stems from the very basic fundamentals of Artificial Intelligence: to model human intelligence. Since the processing of negation by the human brain is more complicated than the processing of affirmation, as

⁶The example question was found in (Cardie et al., 2000), and was first used in the TREC8 QA development corpus.

it was the attempt of chapter 2 to demonstrate, Question Answering systems are similarly expected to require more processing for negative input, rather than the affirmative counter.

First of all, there is not only one answer to the second question. The answer should definitely be a country, but it could also be a number of countries, all the countries in the world that do not have a part of the Amazon rain forest in their maps, just a single country which does not have a part of the Amazon, or even the list of countries in Latin America, which do not have a part of the rain forest.

Either way, all of the above answers are equally valid. How a human being chooses which answer is the appropriate depends on the context of the question: If, for instance, the question is uttered during a geography lesson about Latin America, the answer should consist of Latin American countries and only. Nevertheless, the system would be expected to give the right answer at the right time. It should realize that the answer should consist of countries, but also, take context into consideration before it retrieves the relevant documents.

Secondly, the steps which the system should follow, in order to extract the correct answer to a negative question, the same way as a human would do, are more complicated than the steps needed for answering an affirmative question. What is required for a human (and consequently for the 'optimal system') is:

- to define the context of the negative question;
- to identify the scope of negation; in other words, which part of the sentence is negated;
- to retrieve the documents with the relevant information;
- to reply only with the choice of documents which answer to the question, taking into account the negation (and here is where answering negative questions differs the most from answering affirmative ones).

These steps will be discussed in more detail in the following chapter. The attempt here, was merely to communicate the complexities that Question Answering systems still have to overcome, in order to reach to the point of being able to answer negative questions, with the same speed and accuracy, as they do for positive questions.

3.5 Conclusion

This chapter was an attempt to point out the incompleteness of current Information Retrieval systems as regards the treatment of negative input by the user. Starting from a general overview of IR systems, going to search engines, and finishing with Question Answering systems, which are a specific type of IR, consideration points were raised and discussed, regarding the incorporation of negation in the IR picture.

The next chapter will describe the system requirements for the development of NotFilms, a Question Answering system created for the purposes of this project, which accepts questions about films which were released in 2005. The innovation of NotFilms, is that it is able to read, parse, and correctly answer questions that contain the negative marker 'not'.

Chapter 4

System Requirements

4.1 Question Answering and Databases

According to Diekema et al. (2004), today's restricted-domain Question Answering systems have moved from language understanding to more specialised extraction rules, which make them less dependent on large knowledge bases (Diekema et al., 2004). However, this relationship between database dependence and linguistic analysis is a key issue for the goal of this current project: the development of a Question Answering system which will be able to deal with negation. Since understanding negation requires some linguistic analysis, we will go back to the earlier way of developing restricted-domain systems, and assume a strict dependency on databases. This way, there will be a control over the content of the user's questions, and thus the focus will remain on the linguistic analysis of the input. Hence, the NotFilms system is a restricted-domain system, thoroughly dependent on its knowledge bases, which include information about movies of 2005. It analyses the user's input both in terms of syntax, as well as semantics, before retrieving the answer from the relevant database entries.

4.2 Functional Requirements

The ultimate goal of the present Question Answering system is to be able to correctly answer negative questions. In order to reach this point, there are a number of sub-goals which need to be satisfied:

- a. The system should accept natural language queries, in English, which contain negation.
- b. The system should provide the user with the exact answer.
- c. The system should be extendable to other negative constructions, or other domains.
- d. The system should be able to answer as many questions as possible.
- e. The system should be able to produce an answer within reasonable time.

4.3 CURT Modules Reuse

Since the efforts of the developer are focused on the treatment of negation itself, existing modules, which conduct linguistic analysis, are adopted. For the syntactic and semantic analysis of the input, NotFilms is using modules from a system, called CURT ('Clever Use of Reasoning Tools'), which is available on the internet (Blackburn and Bos, 2005). All the Curt and NotFilms files are written and run in Prolog. An overview of the system can be found at (Blackburn and Bos, 2005). In a few words, Curt is a system which offers interaction with the user in natural language via a dialogue interface. It analyses the user's input in terms of syntax and semantics, using lambda calculus, quantifier storage, model builders and model checking programs, etc., and gives feedback to the user, regarding the consistency of what has been said, and what has been kept in the system's memory, based on the previous inputs by the same user. What are needed from the Curt system in the case of NotFilms, are the modules which will parse the input questions, and give their semantic reading.

What follows is a list of the relevant Curt files, which were incorporated in the NotFilms system. The files used for the NotFilms kept their original names, for extendibility reasons. Additions were made to the first three files (namely `comsemPredicates.pl`, `readLine.pl` and `englishLexicon.pl`):

comsemPredicates.pl This file defines useful Prolog predicates, used for the analysis of the input, as well as the document extraction from the databases. Examples of already defined predicates, used for the development of NotFilms are: `appendLists/3`, `memberList/2`, etc.. Additions have been made, to include predicates such as `removeItems/3`, find common members (`intersection/3`), compare two lists and find members which do not belong to both of the lists (`unintersection/3`), select the first semantic reading, if there is more than one available (`selectFirst/2`), replace items in a list with something else (`replaceFromList/3`), etc.

readLine.pl This file reads the input line and turns the natural language into a format appropriate for linguistic analysis. It turns the capital letters into lower case letters, removes any quotes, eliminates any non-English letters and numbers. Additions have been made, for some initial text pre-processing. These additions will be presented further on, at the presentation of the NotFilms system architecture.

englishLexicon.pl This is where the vocabulary is stored. Apart from the Part of Speech (POS), which each entry belongs to, there is additional information, which will be used for the semantic processing of the entries. All the words include the syntactic entry of the word (the actual realisation of the word), as well as the symbol of the word (the way this word will be read in the semantic representation). In addition, for the quantified NPs (*who* and *what*) and the determiners, there is information about the mood (i.e. declarative or interrogative), and the type (i.e. wh-word, definite or indefinite). For the verbs, there is information about the number of the entry (singular or plural), as well as whether the word is in infinite or finite form. Additions have been made in order to include the database

entries, and as many domain specific verbs in the lexicon as possible. These additions will be presented further on.

englishGrammar.pl This file is responsible for the syntactic parsing of the input. It begins by combining the different words into syntactic phrases, and then combines the phrases into sentences or questions. The semantics of each phrase is passed on to the semantics of the corresponding sentences and questions, where each sentence contains the semantics of all its constituents.

alphaConversion.pl This file is one of the files responsible for the semantic analysis of the input. It involved the use of lambda calculus.

betaConversion.pl Another file involving semantic processes which relate to lambda calculus.

semLexStorage.pl This file uses each word's symbol (found in `englishLexicon.pl`) and associates it with semantic macros, which are stored here.

semRulesKeller.pl This file deals with quantifier scope ambiguities, and thus the realisation of more than one semantic readings, if there is at least one quantifier expression in the sentence. It generates all the appropriate readings for each sentence in a disciplined way.

Further analysis of the use and incorporation of these files into the NotFilms system is done in a subsequent chapter. The following section indicates how the NotFilms knowledge base was created.

4.4 Knowledge Base

The relevant information for the creation of the NoFilms knowledge base was taken from the Internet Movie Database ¹. The files were initially in raw text

¹The Internet Movie Database can be found at: <http://www.imdb.com/>

form.

4.4.1 Information Elicitation

Due to size limitations of Sicstus Prolog, it was not possible to import the full database of over 150MB in the system. For this reason, there was a need to set several specifications as to which films to include in the NotFilms Database. However, this shortcoming will not affect the importance of the present project, since the focus of this question answering system is more qualitative (i.e. to prove that negation can indeed be captured in Question Answering), and less quantitative (i.e. to make large knowledge bases available to the system).

Film Title The database includes only titles which consist of letters from the English alphabet. This excludes titles that have numbers (e.g. 'James Bond 007: From Russia with Love'), non-English letters, or non-alphanumeric characters (e.g. hyphens, punctuation marks, etc.).

Director's Name The database only includes names which consist of letters from the English alphabet.

Actor's Name The database only includes names which consist of letters from the English alphabet.

Actresses No information about actresses is available in the database.

Production Company The database only includes names which consist of letters from the English alphabet.

Year of Release All the films included in the created database have been released in 2005. In general, information about the year of release is essential, in order to differentiate films with the same name. For example, War of the Worlds was first released in 1953, and there was another release this year. Since the title is the same, this additional specification of

year serves to distinguish between the two different films. Although the current database only includes films which have been released in 2005, the year is included, allowing for a smoother future expansion of the database.

Genre The complete Internet Movie Database includes among others the following genres: drama, romance, mystery, adventure, family, musical, comedy, animation, fantasy, science fiction, action, western, crime, thriller, horror, biography, documentary, etc. The genres included in the Not-Films database are only:

1. drama,
2. thriller,
3. action,
4. musical and
5. science fiction.

Language The database includes all (and only) the films which are in English, German and Greek.

Country No constraints were made on the countries in which the films were produced, as long as the language of films was one of the pre-mentioned.

Film Rating Due to the fact that plenty of the 2005 movies have not been released yet, not all the movies have been rated. For this reason, no information about rating was included in the database, although the initial plans were to include it.

Information Completeness Only movies which had all the above information available from the IMDB were included in the database.

The application of these constraints has resulted in the collection of 195 entries of the together/6 predicate, 195 entries of company/3, and 1460 entries of the men/4 predicate.

4.5 Prolog Facts Construction

Using regular expressions in Python, the initial raw text files were transformed into Prolog facts. Three separate sets of predicate facts were used in order to divide the information and thus make the database search less time consuming. The three created predicates had the following format:

1. `together(Film Title, Year of Release, Director, Genre, Language, Country of Filming).`
2. `company(Film Title, Year of Release, Production Company).`
3. `men(Film Title, Year of Release, Actor, Rank of Actor).`

Note that the last argument of the `men/4` predicate (Rank of Actor) is a number. This number indicates the importance of appearance for each actor. A person who starred in a movie (in the example, it is Tom Cruise) will have the number 1, denoting the position of the protagonist. Yet a person who had the smallest role will have a higher number than all the other actors of that particular movie. This information allows the user to ask questions regarding the starring actors.

It is also important to notice that all three predicates include the title of the film, as well as the year of production. This allows the automatic extraction of an answer from only one predicate, without having to match the film title with a particular predicate. To make this point clearer, consider the next two example questions:

1. *Which director did not direct any English movies?*
2. *Which Steven Spielberg movies did Tom Cruise not participate in?*

The answer for the first question can be directly extracted from the predicate `together/6`. All the given and needed information (i.e. director and movies in English) is found there. Yet, the second question requires information from

more than one predicates: the director's name, which is to be found in together/6, as well as the particular actor's name, which is to be found in the predicate men/4. The processing of the second question takes more time, for the very reason that the required data are not included in one predicate. Consequently, to avoid this complexity from at least the 'easier' questions (e.g. the first example question above), and assure quicker processing, the film titles and dates are included in all three predicates.

This is an example of a database entry:

1. together([war,of,the,worlds],[2005],[steven,spielberg],[action],[english],[usa]).
2. company([war,of,the,worlds],[2005],[paramount,pictures]).
3. men([war,of,the,worlds],[2005],[michael,arthur],[28]).
 men([war,of,the,worlds],[2005],[gene,barry],[11]).
 men([war,of,the,worlds],[2005],[david,alan,basche],[12]).
 men([war,of,the,worlds],[2005],[john,michael,bolger],[33]).
 men([war,of,the,worlds],[2005],[clay,bringhurst],[72]).
 men([war,of,the,worlds],[2005],[michael,brownlee],[14]).
 men([war,of,the,worlds],[2005],[art,chudabala],[81]).
 men([war,of,the,worlds],[2005],[benny,ciaramello],[66]).
 men([war,of,the,worlds],[2005],[robert,cicchini],[35]).
 men([war,of,the,worlds],[2005],[tom,cruise],[1]).
 etc., etc.

Storing different kinds of information under the same predicate (as can be seen with together/6) might not be the optimal way of representing data in knowledge bases. It hides high risks of inconsistencies, and forces redundant data entries. For instance, when there are more than one directors for the same film, there are more than one together/6 entries, each time changing only the name of the director. This storing method has created additional difficulties with the initial importing of the whole set of movies into Sicstus Prolog. Since a number of data were stored under the same predicate, the database file became

so huge, that any change (even removal) on the structure of these predicates was extremely time consuming. However, the decision of such an output was made after some initial trials of importing all the information in smaller, separate predicates, as the ones of company/3 and men/4, which had resulted in even slower data processing.

4.6 Domain-Specific Negative Constructions

Unlike the open domain QA systems, which can find ready questions from question logs (AskJeeves, Encarta², etc.), domain specific systems, like the present one, do not have any ready sets of questions in hand (Diekema et al., 2004).

4.6.1 Small Scale Exploratory Study

The way to collect sample questions for NotFilms, was the following: At the very initial stages of the system design, domain specific questions were collected from students. The following email was sent to various student mailing lists of Edinburgh University, asking from students to pose questions that contained *not* or *no*:

I am currently working towards my dissertation and I would like your help. I need to gather 100 questions that contain the negative words 'not' and 'no', to use as my training set for what I am doing.

I am working with some films databases and they offer information about:

- the names of the films*
- the names of the actors*
- the names of the directors*
- the countries the films took place in*
- the years of production*

²*encarta.msn.com*

- the kinds of movies (comedy / science fiction / horror / etc.)
- the language of the films
- the production companies
- the ratings

I need to come up with a big set of negative questions, therefore, if you can think of some (even 2 or 3), it will really help me.

example questions:

1. *Did Steven Spielberg direct any films which were NOT in English?*
2. *Which Japanese thrillers were NOT produced in the 1960s?*
3. *Which country had dramas but not comedies?*
4. *Were there NO documentaries produced by Warner Bros?*

I will be looking forward to your replies.

The complete set of collected questions can be found in Appendix A. The questions are divided between Questions in Passive Voice, questions in Active Voice, and questions containing a copula, or the impersonals 'Is there' and 'Are there' (questions which include more than one verbs might be found in more than one lists). This division shows the three main negative constructions apparent from the questions. The collected questions appear in the Appendix, in order to serve as future question logs, for possible system extension and expansion.

It was indeed hard to collect a number of questions which would contain different constructions, and this might be due to the fact that the email contained example questions. Unfortunately, this only became apparent after the collection of the questions. As can be seen from the given examples of the email, two of the questions were yes/no questions, and the other two were wh-questions. In addition, two were in passive voice and two in active voice. Two of them included the negative determiner *no*, whereas the other two included the adverb *not*. No example question had a simple Subject-Verb-Object (SVO) structure. As a result, the great majority of the collected questions (found in Appendix A) have complicated structures with combinations of positive and negative questions in one query, prepositional phrases, embedded clauses, etc.

4.6.2 Focus on Specific Constructions

Looking at the collected questions, several observations can be indicated. To start with, one student posed the following question: *Is it not true that actors are very rich?* Apart from the fact that either way, there is no available information about the wealth of the actors, still, such a question is pragmatically complex. Such questions are usually rhetorical, and require no answer. I have to give credit to the student who has posed this 'strange' questions, which is helpful enough, as it indicates that this kind of questions are neither processable, nor real.

Another observation is related to the similar distribution of Yes/No questions and Wh-Questions. There are a total of around 50 sentences, half of which are Yes/No questions, and half Wh- questions. This, however, needs more elaboration. The following questions are taken from the set of collected questions:

1. *Has Michael Moore ever directed a film which is not a documentary?*
2. *Were there any actors who played in T.V. movies and did not play in T.V. series?*

Even if a user typed a Yes/No question, s/he might not be satisfied with a Yes/No answer. What the user might want to know in the first example, is which non-documentaries were actually directed by Michael Moore. If the system responded with Yes, then most probably the user would ask *Which movies has Michael Moore directed which are not documentaries?*. The same would be expected for the second example, as well. Therefore, since the Wh-questions require more informative answers, a preference will be given to the treatment of such constructions first, and then of Yes/No constructions.

There was a need to narrow the spam of negative constructions that would be captured by the system. Still, even if one part of these questions was finally captured and correctly answered by the new system, it would be very promising for the expansion of QA systems into capturing more negative constructions. The constructions chosen for the initial phase of the system devel-

opment were active voice questions, which used the negative particle *not*.

The reason why these sentences were chosen was based on which constructions would be correctly parsed by the Curt modules in charge for the linguistic processing, with as little modifications to the existing code as possible. Further elaboration on this will be done in the next chapter, where I am presenting the separate processes of NotFilms.

4.7 Conclusion

This chapter presented the system requirements for the development of NotFilms. It first indicated the functional requirements of the system, and continued with the description of some existing modules which are incorporated in the system. What followed was a description of how the NotFilms databases were built. The chapter finished with the method used for collecting sets of questions relevant to the system, as well as with some specific observations which have impacted the kinds of negation constructions captured by NotFilms. The following chapter presents how the system works.

Chapter 5

Implementation

5.1 System Architecture Components

This chapter is dedicated to the description of the NotFilms system. It presents the three different components of the system, which deal with different functions: the way the system reads negative input which includes the word *not*, the way it processes the information both in terms of linguistic analysis and information retrieval, and finally, the way it makes the answer available to the user. By typing in a question, the user is able to see two distinct pieces of information: first, the semantic reading of the question s/he has just typed, and then, the actual answer to the question.

The general architecture of the NotFilms system can be found in figure 5.1. It denotes the three main components of the system:

Input : It receives the question and pre-processes the text.

Processing : It includes two subprocesses: (a) The linguistic analysis of the input, and (b) Information Retrieval.

Output : It is responsible for putting the answer to the appropriate format, in order to present it to the user.

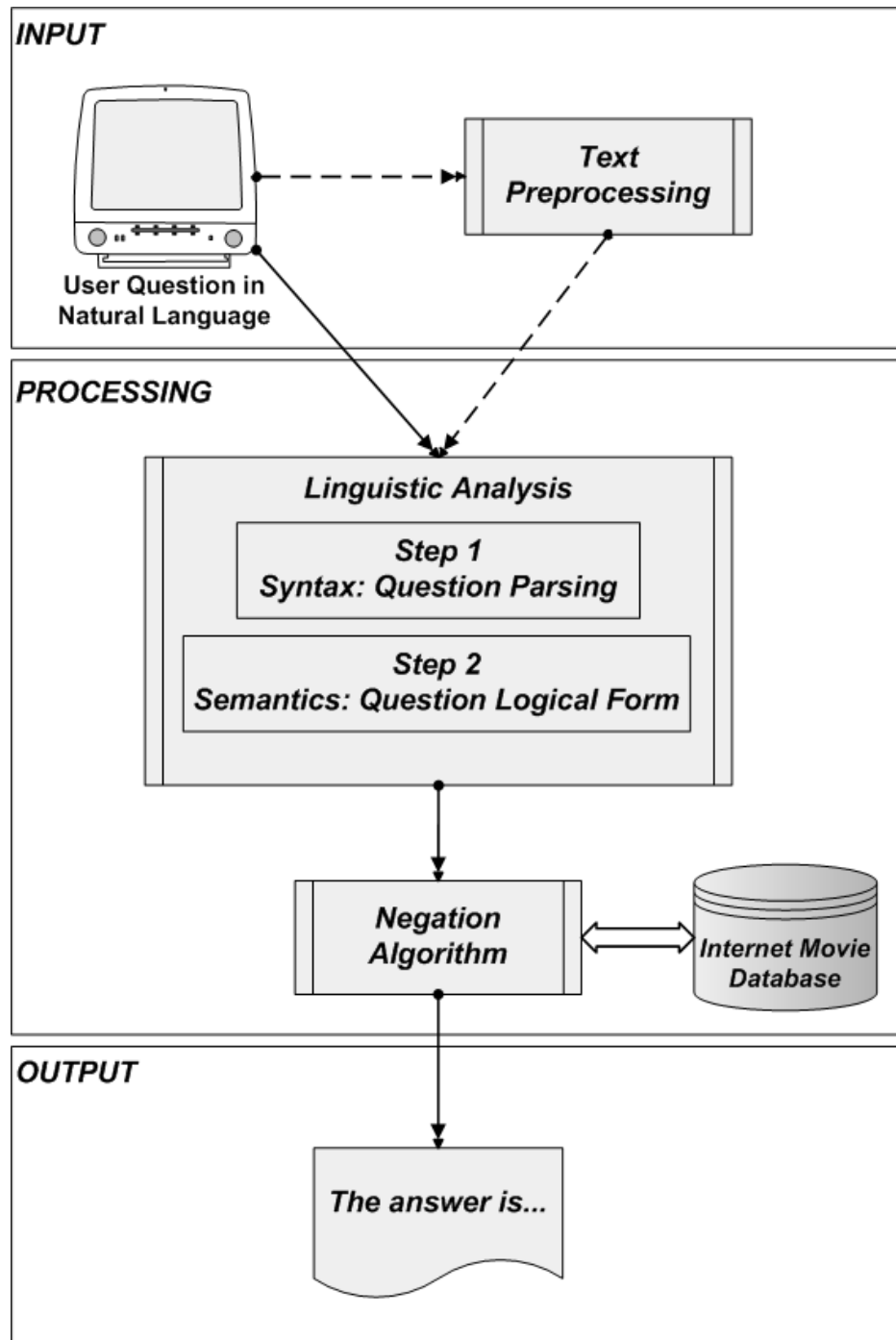


Figure 5.1: System Architecture

5.2 Input Component

The system shows the sign '>' at the beginning of the input line in the Prolog Interface, where the user can type a question in natural language. The kinds of questions accepted by the system depend on whether they can be analysed linguistically by the ready modules of Curt system. We will discuss this in more detail in the next section.

The user's input is filtered (pre-processed) in order to have the appropriate format for the Curt linguistic analysis. All punctuation has been removed, and all letters have turned into lower case, by Curt's Readline module. As an extend, the user can use capital or lower case letters, without affecting the behaviour of the system. Note that due to this process, words like *didn't*, *don't*, etc. become *didnt* and *dont*. This is dealt with in the next section, which focuses on the linguistic analysis.

5.2.1 Move *Not* to the appropriate position

The only way the Curt existing modules realise *not* as a negative marker, and thus are able to give it the appropriate parse, is when it is found exactly after an auxiliary or a copula. For instance, the question '*Which action films does Orlando Bloom not participate in?*' is problematic for Curt, hence, it is not parsed as a grammatical sentence.

This same question can only be accepted, if the negative marker moves next to the auxiliary, i.e. '*Which action films does not Orlando Bloom participate in?*'. However, this is not an acceptable construction in English. Therefore, there needs to be a way to allow the user to type grammatical questions with the correct *Aux + Subject + Verb* construction, which can be accepted by the system.

The suggested solution is as follows: If the word *not* is found, make sure it follows one of the auxiliaries: *do*, *does*, *did*. If it is found in another position in

the sentence, remove *not* from its place and put it next to the auxiliary.

5.2.2 Disambiguate the pronoun *Who*

The only specification of this Wh-pronoun is that it refers to persons. Therefore the system cannot distinguish between the persons of the database, namely actors or directors, resulting in unwanted outputs. An example of a wrong output would be to include all the actors whatsoever together with the correct list of directors, for the question: *'Who didn't direct any science fiction films?'*. Actually, such an output would be a 'valid' answer, considering the fact that all the persons in the database (excluding the ones who did such films) did not actually direct any science fiction movies. However, the user would not expect to find any actors in the answer, and would be displeased with the system's consistency, if the answer included actors.

A solution is needed in order to disambiguate the pronoun, according to the relevant information in the input. An equivalent -unambiguous- question to the one above would be: *'Which directors didn't direct any science fiction films?'*. As an extend, the ambiguity surrounding *Who* is solved by converting *Who* into *Which actors*, if the verb of the sentence is relevant to actors (e.g. *participated, played, starred, appeared*). By the same token, *Who* is converted into *Which directors*, if the verb of the sentence is relevant to directors (i.e. *directed*).

As soon as the above requirements are fulfilled, the input format is ready to be passed to the next component for linguistic processing and interaction with the knowledge base.

5.3 Processing Component: *Linguistic Analysis*

The processing component is divided into two major sub-components:

1. The linguistic analysis of the user's input, and
2. The use of Information Retrieval processes, which involve distinguishing the negated part of the input, if there is any, and interacting with the database until the extraction the exact answer.

As soon as the user's input is pre-processed, the text is passed for linguistic analysis. The ultimate goal for this process is to pass the logical form (the semantic reading) of the sentence to the IR section. This is an example of the logical form of a user's question:

> *Which dramas did William L. Johnson not appear in?*
 [que(X,drama(X),not(appear([william,l,johnson],X)))]

In order to produce the logical form, there are 2 intermediate steps, as in figure 5.1. These steps are explained in detail below.

(A) The Syntax step:

- POS tagging
- Parsing

(B) The Semantics step.

5.3.1 Define the POS of individual words

This takes place in the `englishLexicon.pl` file. Each word of the input has to be accepted as an entry in the lexicon, and receive further specifications. The entries from the database are imported in the lexicon, and are realised as particular Parts of Speech, according to their positions in the database predicates. Here are the entries for **Proper Nouns**, and are explained in the lines below:

```
lexEntry(pn, [symbol:Y, syntax:Y]) :-
    together(Y,_,_,_,_,_);    % FILM
    together(_,_,Y,_,_,_);    % DIRECTOR
    men(_,_,Y,_);              % ACTOR
```

```
company (_,_, Y) .                % COMPANY
```

This can be translated as follows: 'If you find a word Y in the input and can match it with the relevant positions in the database (i.e. if it is a film name, a director's or actor's name, or a production company), treat it as a Proper Noun (PN), and represent it in the reading as Y'.

Regarding the **determiners**, Curt provides information about *every*, *a*, *the* and *which*. The following determiners are added for the purposes of NotFilms:

```
lexEntry(det, [syntax:[an],mood:decl,type:indef]) .    %(an)
lexEntry(det, [syntax:[all],mood:decl,type:indef]) .  %(all)
lexEntry(det, [syntax:[any],mood:decl,type:indef]) .  %(any)
lexEntry(det, [syntax:[],mood:decl,type:indef]) .     %(empty)
```

Determiners are closed-class words and do not relate to the database. All the information needed is stored in the lexicon file. Determiners carry information about the mood (declarative or interrogative), as well as the type (definite or indefinite). The added determiners were 'an' as in *an actor*, 'all' as in *all movies*, 'any' as in *which directors*, and an empty entry, to reason for the absence of determiners in noun phrases, as in the question *Who did not play in [] thrillers?*.

Curt lexicon also includes the two **Quantified Noun Phrases** *who* and *what*. They have similar semantic specifications as determiners, but include a symbol specification as proper noun entries:

```
lexEntry(qnp, [symbol:person, syntax:[who], mood:int, type:wh] ).
lexEntry(qnp, [symbol:thing, syntax:[what], mood:int, type:wh] ).
```

Nouns may directly match to database entries. In such a case, the rule says: If you read a word X which you can match to the genre position of the database, treat it as a noun, and use it as a string Y for the semantic reading. This captures words such as *thriller*, *drama*, *musical*, etc. and the code is given below:

```
lexEntry(noun, [symbol:Y, syntax:X]) :-
```

```

together(_,_,_X,_,_),      %(genre)
concatStrings(X,Y).

```

There is another set of rules which says: If you read the genres in a different form than how they are stored in the database (i.e. *science fiction* instead of *scifi*, or plural forms of genres like *dramas*, *thrillers*, *musicals*), realise them as nouns as well, and use the database version for the semantic reading, in a string format (i.e. removing the surrounding brackets):

```

lexEntry(noun,[symbol:drama,syntax:[dramas]]):-
    together(_,_,_[drama],_,_).
lexEntry(noun,[symbol:thriller,syntax:[thrillers]]):-
    together(_,_,_[thriller],_,_).
lexEntry(noun,[symbol:scifi,syntax:[science,fiction]]):-
    together(_,_,_[scifi],_,_).
lexEntry(noun,[symbol:musical,syntax:[musicals]]):-
    together(_,_,_[musical],_,_).

```

Nouns might also be defined within the lexicon file itself. Several domain-specific words are used within sentences to ask about database entries. The examples below are just a part of the complete list of such nouns. The reading (symbol) of the singular and plural form of the same word is the same, even if the actual realisation of the words in the input differs. For example, *actor* and *actors* use the same symbol for semantic representation, which is the singular one. The complete list of these domain-specific nouns is (giving only the singular form): *language, country, director, production company, company, actor, actress, film, movie, place, genre, type*. Here is how these entries are coded:

```

lexEntry(noun,[symbol:actor,syntax:[actor]]).
lexEntry(noun,[symbol:actor,syntax:[actors]]).
lexEntry(noun,[symbol:film,syntax:[film]]).
lexEntry(noun,[symbol:film,syntax:[films]]).

```


It is important to note that not all the questions which can be formed by these words are actually accepted by the system. However, a bigger list of noun entries was added in the lexicon, for extendibility reasons.

The **noun premodifiers** follow determiners and precede nouns, in the same way as adjectives do. Here is the lexical entry for premodifiers, and is explained in the lines below:

```
lexEntry(adj, [symbol:Y, syntax:X]) :-
    (together(_,_,_,_,X,_);      % (language)
     together(_,_,_,X,_,_);      % (genre)
     together(_,_,X,_,_,_)),      % (director)
    concatStrings(X,Y).
```

The `concatStrings/2` predicate takes a list of items and turns it into a string. For example, it will accept *[steven,spielberg]* and turn it into *stevenspielberg*. Therefore, what the above piece of code says, if the word can be found in the database in the place of language (*English, German, Greek*), genre (*thriller, science fiction*, etc.) in the place of director, or company, you can label it as a premodifier, and use its concatenation for the semantic representation.

Here are examples of each case:

1. *Who did not direct any **English** films?* (language)
2. *Which actor did not appear in **drama** films?* (genre)
3. *Which company didn't produce any **Steven Spielberg** movies?* (director)
4. *Who starred in a **Paramount Pictures** movie?* (production company)

It is important to distinguish between adjectives and pre-modifiers to avoid misunderstandings. The pre-modifiers can belong to different parts of speech (e.g. the directors are actually Proper Nouns), and yet are able to modify - to characterise other nouns. Yet a proper name cannot be seen as an adjective. On the contrary, something whose part of speech is adjective (e.g. *large*), it can only be an adjective and nothing else. Nevertheless, since the pre-modifiers'

position in the sentence is the same as adjectives, and since the grammar and semantic analyses of the Curt files make use of the lexical entry of adjective, the name convention has not changed.

We now proceed to the lexical entries of **transitive verbs**. According to the form of the word realised in the input, the entry includes information about the number (singular or plural), the form (finite or infinitive), as well as the way the verb will be represented in the semantic reading. Additions were made to include domain-specific verbs, as well as the past participle for each individual verb, which was not used in Curt. Here are two examples of transitive verbs, and are explained below:

```
%entry for the verb Produce
```

```
lexEntry(tv, [symbol:produce, syntax:[produce], inf:inf, num:sg]).
lexEntry(tv, [symbol:produce, syntax:[produces], inf:fin, num:sg]).
lexEntry(tv, [symbol:produce, syntax:[produce], inf:fin, num:pl]).
lexEntry(tv, [symbol:produce, syntax:[produced], inf:fin, num:sg]).
lexEntry(tv, [symbol:produce, syntax:[produced], inf:fin, num:pl]).
```

```
%entry for the verb Appear In
```

```
lexEntry(tv, [symbol:appear, syntax:[appear,in], inf:inf, num:sg]).
lexEntry(tv, [symbol:appear, syntax:[appears,in], inf:fin, num:sg]).
lexEntry(tv, [symbol:appear, syntax:[appear,in], inf:fin, num:pl]).
lexEntry(tv, [symbol:appear, syntax:[appeared,in], inf:fin,num:sg]).
lexEntry(tv, [symbol:appear, syntax:[appeared,in], inf:fin,num:pl]).
```

The infinite forms of the verbs appear in questions which include auxiliaries. For example, *Who did not direct German films?*. The finite forms appear in questions which do not contain auxiliaries or negation, like *Which actors starred in science fiction movies?*. The number of the verb is important, because at the parsing stage, the verb has to agree in number with the subject. This way, an ungrammatical question like *Which actors participates in War of the Worlds?* will

correctly fail to be parsed.

Apart from the verb *produce*, the lexicon also includes *direct* and *speak*. Looking at the second verb example, we notice the verb *appear*. One might correctly protest to the classification of such a verb as a transitive verb. Indeed, such a verb is not followed by object nouns, as is expected with transitive verbs. Instead, it is followed by a prepositional phrase. However, since the proceeding preposition is the same at all cases, the structure of this kind of verbs will transit from:

Verb	+	Prepositional Phrase (Prep. + NP)
[appeared]		[in 'Kingdom of Heaven']

to:

(Verb + Adverb)	+	Noun Phrase (Prep. + NP)
[appeared in]		['Kingdom of Heaven']

Treating *in* as an adverb particle attached to the verb, instead of as a preposition particle attached to the following noun phrase, will allow the system to include the particle in the realisation of the verb itself. This **verb-particle construction** has long been discussed in the linguistic fields (Gueron, 1987; den. Dikken, 1995; Wurmbrand, 2000; Ramchand and Svenovius, 2002), etc. By realising *in* within a transitive verb's lexical entry, the system depicts no structural difference between *Who didn't direct thrillers?* and *Who didn't participate in thrillers?* Similar verbs to this are *play in*, *star in*, *participate in*, *take place in* and have been included as lexical entries of transitive verbs.

As we will see later on, NotFilms is able to deal with questions regarding the constituents of transitive verbs (subject and object questions). For this reason, it does not deal with intransitive verbs for the moment. However, a number of intransitive verbs have been included in the lexicon for future consideration. Here is an example:

```
%entry for the verb Take Place
lexEntry(iv, [symbol:takeplace,syntax:[take,place],inf:inf,num:sg]).
```

```

lexEntry(iv, [symbol:takeplace,syntax:[take,place],inf:fin,num:sg]).
lexEntry(iv, [symbol:takeplace,syntax:[take,place],inf:fin,num:pl]).
lexEntry(iv, [symbol:takeplace,syntax:[took,place],inf:fin,num:sg]).
lexEntry(iv, [symbol:takeplace,syntax:[took,place],inf:fin,num:pl]).

```

The lexical entries for **intransitive verbs** include *take place*, *play*, *appear*, *star*. It is important to point out what difference it makes for the same verb to be included in the entries of transitive verbs *take place in* and of intransitive verbs *take place*. The following examples will clarify this:

- a. *Which movie did Bolt Birch star in?*
- b. *Where did Bolt Birch star?*

In example (a), the verb is seen as a transitive verb (*star in*), and consequently the question is about the object of the sentence (the movies). In example (b), the verb is seen as an intransitive one (*star*), and thus the question regards the adjunct - the extra information about the film. Hence, the answer to the first question would be an NP (*Monster Mountain*), whereas the answer to the second question would be a PP (*in Monster Mountain*). However, NotFilms does not reply to adjunct questions at the moment, and the entries of intransitive verbs are waiting for future inclusion in the system.

Copulas and **Auxiliary verbs** have one additional information: polarity. This happens because in Curt, the 'not' particle is incorporated as a feature of auxiliaries and copulas, instead of as a property of verbs themselves. Hence for a question to be positive or negative, it depends on whether negation is featured in the entries of copulas and auxiliaries. The entries for copulas and auxiliaries are as follows:

```

%entry for the Present Tense Copula
lexEntry(cop, [pol:pos, syntax:[is],inf:fin,num:sg]).
lexEntry(cop, [pol:neg, syntax:[is,not],inf:fin,num:sg]).
lexEntry(cop, [pol:pos, syntax:[are],inf:fin,num:pl]).

```

```
lexEntry(cop, [pol:neg, syntax:[are,not],inf:fin,num:pl]).
```

```
%entry for the Past Auxiliary
```

```
lexEntry(av, [syntax:[did],inf:fin,num:sg, pol:pos]).
```

```
lexEntry(av, [syntax:[did,not],inf:fin,num:sg, pol:neg]).
```

```
lexEntry(av, [syntax:[didnt],inf:fin,num:sg, pol:neg]).
```

```
lexEntry(av, [syntax:[did],inf:fin,num:pl, pol:pos]).
```

```
lexEntry(av, [syntax:[did,not],inf:fin,num:pl, pol:neg]).
```

```
lexEntry(av, [syntax:[didnt],inf:fin,num:pl, pol:neg]).
```

Here, we can depict the polarity information which is either negative or positive (*pol:neg* or *pol:pos*). This information is needed in order for the semantic processes to produce a negative or positive reading for the questions accordingly.

In addition, notice that there exists a *didnt* entry in the auxiliaries. This has been added for NotFilms. Remember that in the text pre-processing of the system, the apostrophes were removed. This means that in case there is a *didn't* or a *doesn't*, etc. in the input, it will be transformed into *didnt* and *doesnt* respectively. Therefore, if at the present stage, the system reads a *didnt* or *doesnt* in the sentence, it should realise it as negative, and give it the appropriate specifications.

Note that the Curt files only include the entries of the present tense copula, and the singular form of the auxiliary. Therefore, the lexicon has been extended to include the past tense copulas (*were*, *were not*, *was* and *was not*), and the plural forms of the auxiliary (*do*, *do not*, *dont*, *did*, *did not* and *didnt*).

NotFilms does not make use of any **prepositions** for the moment, since the questions it accepts are active voice subject and object questions. However, Curt includes entries for the prepositions *about*, *in*, *of* and *with*, and the entries *at* and *by* have been added for future consideration (e.g. when NotFilms is to accept passive voice questions as well).

Finally, two **coordinators** *and* and *or* are provided by the Curt lexicon with the following entries:

```
lexEntry( coord, [syntax:[and], type:conj] ).
lexEntry( coord, [syntax:[or], type:disj] ).
```

These lexical entries include the coordination type for each word (conjunctive for *and* and disjunctive for *or*) and will influence the semantic representation of questions which contain such coordinations.

The pages above have demonstrated the format of the system's lexicon, and the way the lexical entries are shown in the file. When the Parts of Speech of the words are decided, they are received by the grammar section, which is in charge of parsing the tree structure of the input question.

5.3.2 Parse the Question

Curt's `englishGrammar.pl` file is in charge of combining different parts of speech into phrases, and then from phrases to sentences or questions. There has been no change and no addition to this file. Nonetheless, I will briefly go through the combinations it accepts in order to create phrases and then sentences, which will mirror the kinds of questions acceptable by NotFilms.

An input can either be realised as a sentence, or a question. Needless to say, in order for NotFilms to produce a reply, there needs to be at least one variable in the user's input, therefore only questions are relevant to the system.

A noun phrase NP can be:

- NP + conjunction (*and/or*) + NP
- [] (empty) e.g *Which films did Tom Cruise star in []?*
- Determiner + Noun
- Proper Noun
- Wh- NP

A Wh-NP can be:

- A Quantified NP by itself (i.e. *who* or *what*)
- A Determiner with an interrogative mood (i.e. *which*) + noun

A Noun can be:

- Noun + conjunction (*and/or*) + Noun
- Adjective (or pre-modifier) + Noun
- Noun
- Noun + Noun Modifiers (e.g. PPs and Relative Clauses)

However, since NotFilms accepts questions regarding transitive verb constituents, it does not make use of this latter choice.

A verb phrase VP can be:

- VP + conjunction + VP
- Auxiliary + VP
- Copula + NP
- Intransitive Verb
- Transitive Verb + NP
- Ditransitive Verb + NP + NP

Note here, that NotFilms does not make use of intransitive or ditransitive verbs. In addition, due to the fact that a copula can only be followed by a noun phrase, the system can parse neither questions with copulas which are followed by PPs (e.g. *Which films are not in English?*), nor questions with the verb *to be*, followed by the past participle (i.e. questions in the passive).

Having visited Curt's syntactic phrases, it is time for the root rules, which define whether an input is a sentence or a question. Since the role of NotFilms is to provide information by answering user's questions, only the questions structure is relevant to this project.

Hence a question is realised as:

- WHNP + VP (e.g. *Who did not direct science fiction films?*)
- WHNP + SINV (*Which dramas did Warner Bros produce []?*)

And the only relevant sentence rule is the inverted sentence SINV:

- Auxiliary + NP + Gapped VP
 (e.g. *didn't Orlando Bloom star in [GAP]*)

Once the parsing of the question is ready, and the features from individuals parts of speech have been passed on to the question level, the semantic reading of the question can be produced.

5.3.3 Produce the Semantic Reading of the Question

The Curt files used for this purpose are the `alphaConversion.pl`, `betaConversion.pl`, `semLexStorage.pl` and `semRulesKeller.pl`, which find the specifications they need from the lexicon and grammar files, and produce the semantic reading of the parsed question. No change has been made to the semantic files of Curt.

In case there are more than one semantic readings (depending on the number of quantifiers in the input), `NotFilms` is specified into accepting the first reading on the list of readings which Curt offers, and treat it as the only reading of the particular question. This way, if there is just one reading for each syntactic structure, the information retrieval processes can more efficiently use the semantic pattern of this structure to retrieve answers, and capture as many example inputs with the same structure as possible.

One example of a semantic reading is the following. Note the existence of two quantifiers in the sentence: *which* and *any*. However, only one reading is enough for the `NotFilms` semantic analysis:

```
> Which actors did not star in any movie?
[que( X,actor(X),not(some(Y,and(movie(Y),star( X, Y)))))]
```

The linguistic processing of the input is now complete. The semantic reading of the question is passed on to the Information Processing Algorithms in order to retrieve the answer. Up to this point, the system has received the input,

pre-processed it, produced its semantic readings, chose the first reading, and is now called to process the information by communicating with the database. This step is illustrated in the following section.

5.4 Processing Component: *The Negation Algorithm and IR Processes*

All the IR processes are based on the linguistic interpretation of the input, hence the semantic reading of each question is the number one requirement for this step. The higher code which brings all this together, is the following:

```
curtTalk(run):-
    readLine(Input),
    curtUpdate(Input,_,State,ReadingsAll),
    selectFirst(ReadingsAll,Reading),
    write(Reading),nl,
    process(Reading),
    curtTalk(State).
```

The `curtTalk/1` predicate sets off the beginning of the system. The input is received for text pre-processing by the `Readline` module, and is then passed for the production of the semantic readings with the predicate `curtUpdate/4`. In case the input cannot be linguistically analysed (i.e. the user writes unintelligible words, spelling errors, or a structure that cannot be parsed by the Curt files), then no semantic reading is produced and the IR processes do not need to go any further. The user gets a respond that his/her input could not be processed (see figure 5.2 for example). Otherwise, if there are semantic readings for the input, the system chooses the first one with the predicate `selectFirst/2` and writes it on the screen. This is a way for the user to be notified that the system is processing the input, instead of having silence until the final production

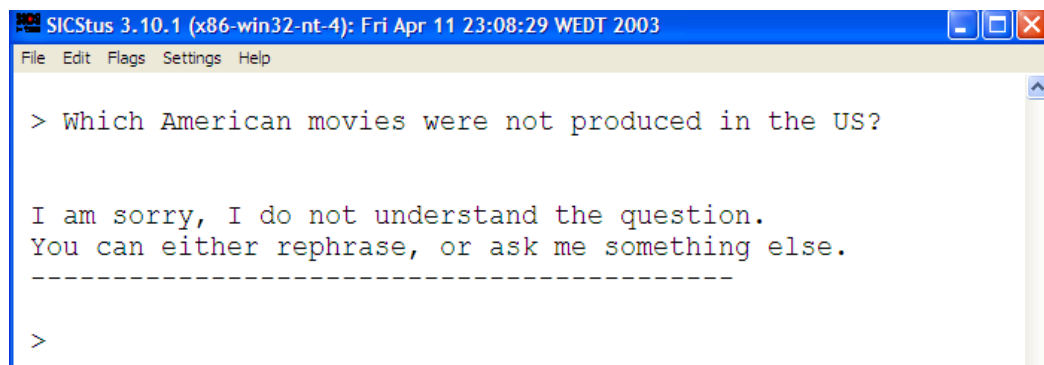


Figure 5.2: The system's reply when the Passive Voice input cannot be parsed

of the actual answer.

The Process module reads the semantic reading which is given as a list, and decomposes this list, to *find* its crucial components that will lead to the answer of the question. The AlgorithmIR predicate is crucial for the IR process. It joins together the IR processes of the separate parts of the semantic reading, and produces the final answer. Note that the write2/1 predicate is defined with the way the answer is going to be presented to the user. Therefore, it belongs to the Output process, which is discussed in the following section. This is how the Process module is defined:

```
process(Reading):-
    algorithmIR(Reading,Answer),
    write2(Answer).
```

The first argument of algorithmIR/2 is a list containing the semantic reading of the input. This list includes one predicate que/3:

```
[ que(X,    lambda expression of X,    binding operators) ]
```

where *que* refers to 'question', *X* is a variable, the lambda expression specifies *X*, and the binding operators give the extra information which needs to be taken into consideration in order for the question to hold in semantic terms.

Here are some examples of questions along with their readings (produced by the system):

- 1a. *Which actors played in Lost Girls?*
- 1b. `[que (X, actor(X), play(X,[lost,girls]))]`

- 2a. *Which English drama movie did Cary Cremidas direct?*
- 2b. `[que(X, and(english(X),and(drama(X),movie(X))),
direct([cary,cremidas],X))]`

- 3a. *Who didn't direct science fiction films?*
- 3b. `[que(X, director(X),
not(some(Y,and(and(scifi(Y),film(Y)),direct(X,Y)))))]`

5.4.1 The Negation Algorithm

```
algorithmIR([que(_, Specifications, Bindings)], Output):-
    find(Specifications, Out1, Negation),
    find(Bindings, Out2, Negation),
    findanswer(Negation, Out1, Out2, Output).

findanswer(Negation, Out1, Out2, Output):-
    Negation==1,    unintersection(Out1, Out2, Output).

findanswer(_, Out1, Out2, Output):-
    intersection(Out1, Out2, Output).

find(not(Negation), Out, 1):-
    find(Negation, Out, 1).
```

The `algorithmIR/2` predicate asks from `find/3` to create a list with all the relevant entries for `Specifications` and another list with the relevant entries of

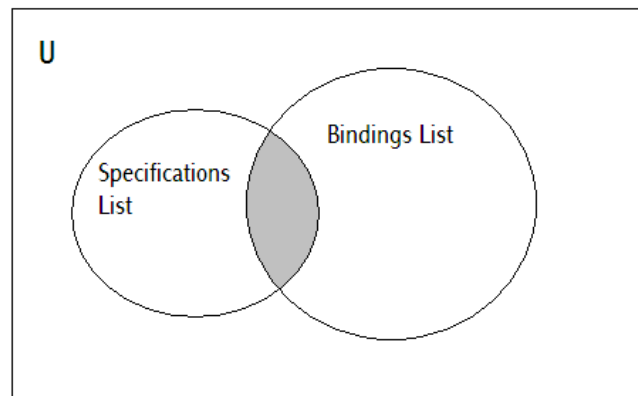


Figure 5.3: The Intersection of Specifications and Bindings Lists

Bindings. The two lists are then combined to reach the final answer. The way they are combined depends on the polarity of the question.

When the input does not include negation, it means that the polarity of the question is positive. In this case, the algorithm calculates the intersection of the two lists. According to Set Theory, the intersection of two lists is $A \cap B$, and is represented in First Order Logic with $\{x \mid x \in A \wedge x \in B\}$ (see figure 5.3).

Note that if there is negation in the sentence, it appears within the Binding Operators. The system recognises the existence of negation, by instantiating the Negation flag. This flag remains uninstantiated, unless there is a *not* anywhere in the reading. In such a case, the system chooses the members of the Specifications List, which are not included in the Bindings List to be the final output. Set Theory defines this process as $A \setminus B$. First Order Logic represents it as $\{x \mid x \in A \wedge x \notin B\}$ (see figure 5.4).

To illustrate the different processes according to the polarity, consider the following two questions. The first question has a positive polarity and thus requires an intersection process.

1. Which dramas did BBC produce?

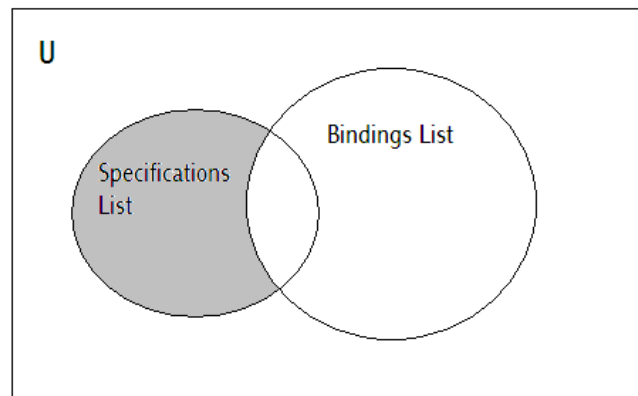


Figure 5.4: The Difference process for the Negation

For this question, the Specifications list includes every entry in the database which satisfies the property of being a drama, that is *all* drama movies. The Bindings list consists of every movie which was produced by BBC. The intersection of these two lists is the set of drama movies which BBC produced.

The second question has a negative polarity:

2. *Which dramas didn't BBC produce?*

The two created lists for this question are exactly the same as the ones of the previous example. Nevertheless, due to the negative polarity of the sentence, a different process is required: The dramas which appear in the BBC set are removed from the complete set of dramas, and what is left constitutes the answer.

5.4.2 Retrieving Information with FIND/3

We now proceed to see in practice how the Find module decomposes the semantic reading into finding the relevant expressions for the Specifications and Bindings part, and how it accesses the knowledge base into retrieving the relevant lists.

NP Resolution

Remember that both the Specifications and the Bindings can have complex readings, according to the amount of information given. For instance, if an NP is a simple noun, it has a relatively small reading. The more pre-modifiers it gets, the more complex its reading becomes. Below we see how the semantic reading of different NPs looks like. Notice that the bigger the NP is, the longer reading it has:

- 1 *a movie* \rightarrow $\text{some}(X, \text{movie}(X))$
- 2 *a thriller movie* \rightarrow $\text{some}(X, \text{and}(\text{thriller}(X), \text{movie}(X)))$
- 3 *a German thriller movie* \rightarrow
 $\text{some}(X, \text{and}(\text{and}(\text{german}(X), \text{and}(\text{thriller}(X), \text{movie}(X))))$

The quantifier readings *some(X)* and *all(X)*, which are usually found at the beginning of the NP readings, are ignored. This happens so, because the system retrieves all the database entries relevant to a question by default. If there are more than one entries in the database that can be included in the answer, all will be available to the user.

In general, the way the Find module deals with complex Noun Phrases is as follows:

First step Decompose the NP readings into their smallest parts (e.g. looking at the 3rd example above, depict the *german(X)*, *thriller(X)*, and *movie(X)* parts)

Second Step Take the head predicates (e.g. *german*, *thriller*, *movie*) and search the database to see if you can find them as arguments. For example, *german* can be found in the predicate:

together([Title],[Year],[Director],[Genre],[german],[Country]).

If there is no entry for a word, this means that the word belongs to the

pre-defined words in the lexicon, therefore do the following:

- a) If the head predicate is *director*, *production company* or *actor*, find all the entries of directors, companies and actors respectively
- b) If the head predicate is *film*, *films*, *movie* or *movies*, automatically find all the film entries in the database.
- c) If the head predicate is *thing*, which is how the wh-pronoun *What* translates into, then treat it as referring to *film*, and proceed in the same way as above.

Third step Create separate lists, each satisfying a particular specification (for our example: one list includes the [Names and Dates] of German movies, another the [Names and Dates] of thrillers, and the third one the [Names and Dates] of all the movies).

Fourth step Find the intersection of all the three lists and put them in a new list. This last list is the output for the whole noun phrase.

VP Resolution

The most focal information in a sentence is the verb. It is seen as the head of the sentence, and thus in semantics, it appears in a predicate position. The verb's constituents are semantically represented as its arguments. Intransitive verbs include one argument (the subject), transitive verbs include two arguments (first the subject and then the object) and so on.

The semantic reading of the input question first defines the properties of the NP constituents, and finishes with the verb predicate. Within the verb predicate, one can depict the verb of the question, as well as which NP is in the subject position, and which one is in the object position.

In the previous section, I showed how the Find module decomposes the semantics of the NP, in order to retrieve the appropriate entries. Since the system

is called to retrieve entries in order to instantiate variables, this means that the particular NP is what the user asks about.

Subject or object questions include verb predicates, of which one constituent is given- known, and the other is unknown. Thus a subject question implies an unknown variable in the subject position, whereas an object question implies the unknown variable in the object position. This is depicted within the verb predicate of the logical form:

Subject Q: *Who starred in 'Stay'?*

[que(X,actor(X),star(X,[stay]))]

Object Q: *Which movie did Ewan McGregor star in?*

[que(X,movie(X),star([ewan,mcgregor],X))]

Already the NP resolution process has produced the list which includes the relevant entries (a list of all the actors for the subject question, and a list of all [Movies: Dates] for the object question). The VP resolution process is called to match the given knowledge with the unknown, taking into consideration the kind of verb which is used. When saying 'given knowledge', I mean the verb and the subject, in the case of object questions, and the verb with the object, in case of subject questions.

In the process of the Binding decomposition, the last element to be defined is the verb predicate, for example *direct(X,Y)*. In this case, either X or Y is known, and thus not an uninstantiated variable, and the system needs to find the other variable. Depending on the syntactic position of the unknown constituent, there are separate processes, which are discussed below.

Unknown Object

The system finds a verb predicate with the first argument instantiated, and the second one uninstantiated (e.g. *star([ewan,mcgregor],X)*). When the unknown variable is the object, then we know for sure that it must become instantiated with either a film title, or a list of film titles. Besides, all verbs in the database

take movies as their objects.

Depending on the verb, the system searches the database, and automatically stores all the subject's matching movies in the output list. To clarify things, this is one example code:

```
find(star(X,Y),Out,_):-
    nonvar(X),
    findall([Title:Date],
    men(Title,[Date],X,[1]),
    Out).
```

This says that in case the uninstantiated variable of the verb *star* is the object Y, find all the film titles along with their dates, in which the subject X starred, and keep them in the list. If the verb regarded directors (i.e. *direct*), then the only difference would be that the search would be taking place within the together/6 predicate, which stores the directors.

Unknown Subject

The system finds a verb predicate, whose first argument is an uninstantiated variable, and second variable is instantiated (e.g. *star(X,[stay])*). This requires more processing steps, for the reason that depending on the verb, the subject could be a list of actors, directors, or production companies. In addition, the instantiated variable, i.e. the known object, could be just a film title, like in the given example, or a number of specifications (e.g. 'English action films'), which have already been resolved by the NP resolution processes, producing a list in the format of [Title:Date].

In case the object is a Film Title, like in the example *Which company produced 'Gods of Los Angeles?'*, the output is directly found from the database. According to the verb of the question, the relevant database predicate is searched. If, for example, the verb is *produce*, then the subject can only be a production company, thus the only place the system will look for an answer is within the predi-

cate company/3. The name of the production company X which has produced this known film title Y is found within the database predicate *company*(Y,_,X). The subject X is thus kept in the output list.

However, recall that the NP processes also deal with the resolution of complex NPs. Such a noun phrase could appear in the subject position, like for example *Which director didn't direct any English action films?*, where the verb in question is *direct*(Unknown Subject, [A list of English Action Movies]). Remember that the NP resolution processes have already dealt with this complex NP, and have put all the relevant database entries in a list which has the format [Film Title: Date]. Again depending on the verb, the search takes place within a particular database predicate, and the appropriate subjects are retrieved. In the given example, the verb *direct* prompts the system to search directly within the *together* predicate. This is the relevant code:

```
find(direct(X,Y),Out,_):-
    nonvar(Y),var(X), finddirectors(Y,Out2),
    removeDuplicates(Out2,Out).
finddirectors([],[]).
finddirectors([H|T],[Director|Out]):-
    [Title:Date] =H,
    together(Title,[Date],Director,_,_,_),
    finddirectors(T,Out).
```

In a few words, the already instantiated object, which is a list of film titles along with their date, is called with a new predicate. This new predicate decomposes the object list, searches the *together*/4 predicate to find the entries of these films, finds the directors of each of these films, and keeps all the retrieved directors in the output list.

Questions with Coordinations

It has been mentioned that the system does not deal with complex constructions with embedded clauses or prepositional phrases, and only deals with simple subject/object questions. However, since the Curt modules do parse questions with coordinations, the decision was to expand the coverage of the system by including the parsable coordination constructions within subject questions, like:

Who didn't direct German films or didn't direct English films?

Which actor played in action films and didn't play in thrillers?

Which actor didn't play in a Steven Spielberg film or a Gino Cabanas film, or a Breck Eisner movie?

Which company did not produce 'Sacrifice' and produced musicals?

Multiple phrase are coordinated either with *and* or with *or*. The Find module is used here as well, to retrieve entries which would answer to this kind of questions. Hence all the positive/negative combinations of verbs are allowed, as shown in figure 5.5: positive VP with positive VP, positive with negative, and also the other way round.

In three of the four possible combinations (positive+negative, negative+positive, negative+negative), the two coordinators are dealt with the same process:

Negative AND Negative Example: *Who didn't play in English movies and didn't play in German movies?* Two relevant lists are created, the first one including a list of all the English movies, the second including a list of German movies. The two lists are combined to create a third list. The answer is the list of actors who did not participate in any of the films included in the third list.

Negative AND Positive Example: *Which companies did not produce any English movies or produced German movies?* Two relevant lists are created, one with all the companies that did not produce any English movies, and

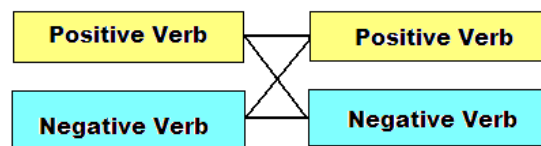


Figure 5.5: Combinations of coordinated verb structures

one with the companies which produced Greek movies. The difference between the two lists is calculated (removing the common members of the two lists) and what has remained in the second list (of companies which produced Greek films) is the answer.

Positive AND Negative Example: *Which director directed Greek movies and didn't direct dramas?* The same process as the one above, with the only difference that after the common members of the two lists are removed, what is included in the first list, this time (i.e. the list of people who directed Greek movies, excluding the ones who directed dramas) is the answer.

The only case where there is a different process for each coordinator is the following:

Positive AND Positive Example: *Which actors played musicals and played Greek films?* The list which contains the answer consists of the intersection of the two separate lists (i.e. the common members of the list of actors who played in musicals and the list of actors who played in a Greek movie). Thus all the actors who appear in the answer must have appeared both in musicals and in Greek movies.

Positive OR Positive Example: *Which actors played in musicals or played in Greek films?* The list which contains the answer is the combination of the two separate lists. The members of one list and the members of the other are joined together and are all included in the answer. Thus it is not necessary for all the actors in the answer to have played in musicals AND in Greek movies. As long as an actor has played in one musical, or one

Greek movie, then he can included in the answer.

Besides, this distinction between AND and OR stems from their logical interpretation. In logic, the relation $A \text{ AND } B$ is realised as $A \wedge B$, and holds true, only if both A and B are true. On the other hand, the relation $A \text{ OR } B$ is realised as $A \vee B$ and holds true, if at least one of the members A and B is true. This means that even if A or B does not hold true, the condition can still succeed.

More examples of coordinated structures are given in the following chapter.

5.5 Output Component

The questions that the system accepts are short answer questions. The exact answer is directly given to the user in the form of

'The answer is: [...]'.

The answer has been kept in a list, within the AlgorithmIR module, and consists of either names of People (the directors or actors), names of production companies, or names of Movies along with their year of release, in the format [Title: Year]. In case of no answer (i.e. no matching entries from the database), the list is empty.

The module Write2 checks whether an answer has been kept in the AlgorithmIR module, and thus it shows it to the user. Otherwise, if the list is empty, it informs the user that no matches have been found:

```
process(Reading):-
    algorithmIR(Reading,Answer),
    write2(Answer).
```

```
write2([]):-
```

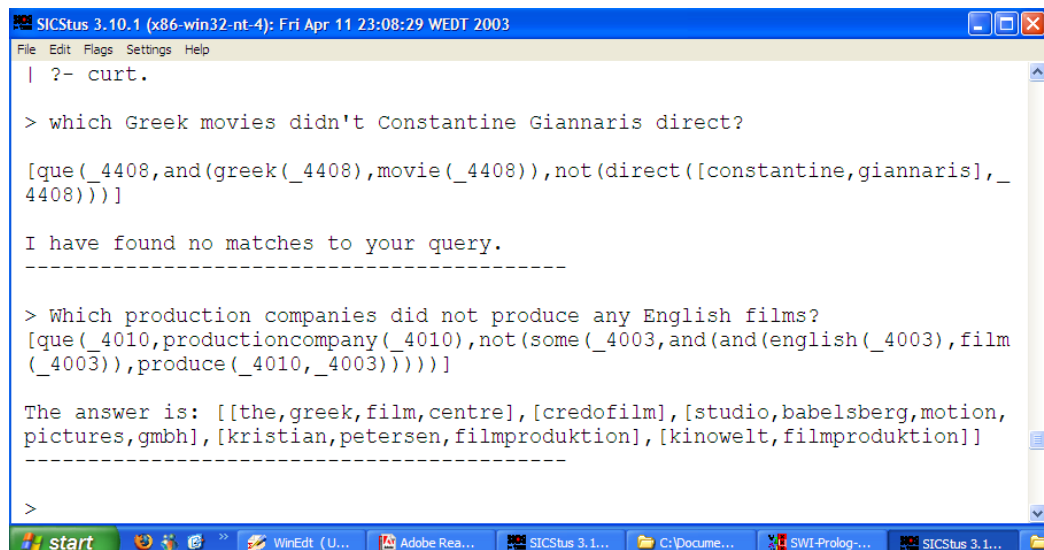


Figure 5.6: Multiple or Zero Results

```
nl,write('I have found no matches to your query. '),nl,
write('-----'),nl.

write2(Answer):-
    nl,write('The answer is: '), write(Answer),nl,
    write('-----'),nl.
```

In general, there is no assumption of having one existing answer for each question. There might be the case that a question gives no results, or multiple results. In figure 5.5 we can see relevant examples. In the first case, only one Greek movie was released in 2005, and was directed by the specified director. Consequently, the answer for the question is zero. In the second case, the answer includes a number of production companies, which are presented in a list.

5.6 Conclusion

In this chapter, I have navigated the reader through the processes of NotFilms, showing how the system is implemented. The implementation of the system takes place with the use of three main components: The Input Component, which reads and pre-processes the natural language question, the Processing Component, which first produces the semantic reading of the sentence, and then calls the retrieval processes in order to extract the answer, and the Output Component, which controls the way the answer is presented to the user.

The kinds of questions accepted by the system are subject and object questions. More than one verbs are allowed in subject questions, with the use of coordination. The crucial contribution of the system is that with the application of its processes, it accepts and analyses both positive and negative questions with the same efficiency.

The next chapter includes examples of natural language input, which is correctly dealt with by NotFilms.

Chapter 6

Examples

It is time to show some real hands-on examples of the system. As can be recalled, NotFilms is able to accept domain specific subject and object questions. The system processes the range of positive questions acceptable by the system, along with all their negative counterparts with an equal efficiency. Negative questions, even with a narrow sense of the term *negation*, have become accepted by the NotFilms QA system.

The following examples cover the structures of questions manageable by the system. They are divided between Subject Questions, Object Questions, and Questions which include coordinations. The natural language input, the semantic reading, as well as the retrieved answer, are shown the same way the system presents them to the user. However, in cases where the answer contains a list of more than 4 lines, the answer is not included in the chapter. Comments on questions are written in italics.

6.1 Subject Questions

- Who directed 'war of the worlds'?

[que(X,director(X),direct(X,[war,of,the,worlds]))]

The answer is:[[steven,spielberg]]

Remember here, that the Input component removes all punctuation from the given question, before it passes it to the Linguistic component for further processing.

- **Who starred in a Steven Spielberg movie?**

[que(X,actor(X),some(Y,and(and(stevenspielberg(Y),movie(Y)),star(X,Y))))]

The answer is: [[tom,cruise]]

The name of a director can appear in a pre-modifier position, as well as in the subject position.

- **Who didn't direct any English movies?**

[que(X,director(X),not(some(Y,and(and(english(Y),movie(Y)),direct(X,Y)))))]

The answer is:

[[natalie,boyatt],[joey,mcadams],[christian,schoyen],
[greg,zeschuk],[constantine,giannaris],[mirko,borscht],
[michael,brynntrup],[andreas,dresen],[juana,dubiel]...

- **which production companies did not produce dramas?**

[que(X,productioncompany(X),not(some(Y,and(drama(Y),produce(X,Y)))))]

The answer is:

[[handheld,films],[crave,films],[anthill,productions,llc],
[new,world,pictures]...

Notice that nouns can be preceded with zero determiners (to account for the case of the indefinite Plural).

- **which actors starred in English science fiction films?**

```
[que(X,actor(X),some(Y,and(and(english(Y),and(scifi(Y),
film(Y))),star(X,Y))))]
```

The answer is:

```
[[bolt,birch],[vincent,ventresca],[ezekel,cruz]]
```

Remember that there can appear more than one pre-modifiers. In this case, there are two: language and genre premodifiers.

- **which actors did not participate in any Paramount Pictures movie?**

```
[que(X,actor(X),not(some(Y,and(and(paramountpictures(Y),
movie(Y)), participate(X,Y)))))]
```

The answer is:

```
[[brian,patrick,clarke],[vernon,forrest],[game],
[houston,summers],[paul,bubbo],[nino,dicosola],
[jason,alan,erickson],[joe,estevez],...
```

Production companies, as well as names of directors, can appear in a noun pre-modifier position.

- **which directors directed German action movies?**

```
[que(X,director(X),some(Y,and(and(german(Y),and(action(Y),
movie(Y))), directX,Y))))]
```

The answer is: [[thomas,frydetski]]

6.2 Object Questions

- **which Steven Spielberg movie did Tom Cruise star in?**

[que(X,and(stevenspielberg(X),movie(X)),star([tom,cruise]X))]

The answer is: [[[war,of,the,worlds]:2005]]

This is a case of verb-object particle construction, where the particle 'in' is treated as part of the transitive verb 'star in'.

- **Which German movies did Rainer Baumann not participate in?**

[que(X,and(german(X),movie(X)),
not(participate([rainer,baumann],X)))]

The answer is:

[[[kombat,sechzehn]:2005],[[willenbrock]:2005],
[[max,und,moritz,reloaded]:2005]]

Be reminded that in order to allow for the auxiliary and the negative marker to be divided, the Input component pre-processes the question by moving 'not' next to the auxiliary, before it passes it to the Linguistic component.

- **What did 'Morning Star Pictures' produce?**

[que(X,thing(X),produce([morning,star,pictures],X))]

The answer is: [[[in,my,sleep]:2005]]

- **Which Greek drama movie did the greek film centre produce?**

[que(X,and(greek(X),and(drama(X),movie(X))),
produce([the,greek,film,centre],X))]

The answer is: [[[omiros]:2005]]

- **Which Paved Productions movie does Jordan Ellis direct?**

[que(X,and(pavedproductions(X),movie(X)), direct([jordan,ellis],X))]

The answer is: [[[guns,before,butter]:2005]]

- **Which Paved Productions movie did Jordan Ellis not direct?**

[que(X,and(pavedproductions(X),movie(X)),

not(direct([jordan,ellis],X)))]

I have found no matches to your query.

The system can produce the semantic representation of sentences in both the Present Tense and the Past Tense.

6.3 Questions with Coordinations

- **Which directors directed science fiction films or directed musicals?**

[que(X,director(X),or(some(Y,and(and(scifi(Y),film(Y)),
direct(X,Y))),some(Z,and(musical(Z),direct(X,Z)))))))]

The answer is:

[[alexander,alcarese],[michael,baumgarten],[chris,columbus],
[tim,cox],[ted,rivera],[eric,wostenberg]]

The answer includes directors who directed at least one of the two mentioned genres.

- **Which directors directed science fiction films and directed musicals?**

[que(X,director(X),and(some(Y,and(and(scifi(Y),film(Y)),
direct(X,Y))),some(Z,and(musical(Z),direct(X,Z)))))))]

I have found no matches to your query.

The answer would only include directors who directed both genres of films.

- **which actors did not play in Black Dawn or War of the Worlds or Kingdom of Heaven?**

```
[que(X,actor(X),not(or(play(X,[black,dawn]),
or(play(X,[war,of,the,worlds]),play(X,[kingdom,of,heaven]))))))]
```

The answer is:

```
[[brian,patrick,clarke],[vernon,forrest],[game],
[houston,summers],[paul,bubbo],[nino,dicosola],...
```

The answer includes actors who did not play in either of the pre-mentioned films.

- **who did not play in musicals or did not play in science fiction movies?**

```
[que(X,actor(X),or(not(some(Y,and(musical(Y),play(X,Y)))),
not(some(Z,and(and(scifi(Z),movie(Z)),play(X,Z)))))))]
```

The answer is:

```
[[brian,patrick,clarke],[vernon,forrest],[game],
[houston,summers],[matthew,goode],[michael,arata],
[laurence,fishburne],[jumper,lark],...
```

The answer includes actors who did not play in musicals, and actors who did not play in science fiction movies, even if some actors did play in the other genre.

- **Which actors did not play in Batman Begins and played in Kingdom of Heaven?**

```
[que(X,actor(X),and(not(play(X,[batman,begins])),
play(X,[kingdom,of,heaven])))]
```

The answer is:

```
[[jouko,ahola],[shane,attwooll],[orlando,bloom],
```

[peter,cant], [marton,csokas],[emilio,doorgasingh],...

- **Which production companies didn't produce dramas and produced action films?**

[que(X,productioncompany(X),and(not(some(Y,and(drama(Y),
produce(X,Y)))), some(Z,and(and(action(Z),film(Z)),
produce(X,Z)))))))]

The answer is:

[[crave,films],[di,bonaventura,pictures],[herold,productions],
[gorilla,pictures],[el,norte,productions],[porter,productions]...

6.4 Conclusion

This chapter has exhibited the kinds of structures and lexical entries which the system can accept. The following chapter discusses on the pros and cons of NotFilms, by providing an evaluation of the current system.

Chapter 7

System Evaluation

There are many suggestions (Nyberg and Mitamura, 2002; Fukumoto et al., 2003; Diekema et al., 2004), etc. with different approaches as to how open domain and closed domain QA systems should be evaluated. Nevertheless, the aim of evaluation in current years is more straight forward; that is, to promote a more human-like information retrieval: one that would provide an answer in natural language, instead of a ranked list of documents where the answer can be retrieved by the user.

Besides, in real life, when a person is expected to provide an answer, s/he does not answer with a list of, say, sentence-like occurrences of a possible answer stored in his/her mind. Instead, s/he makes the appropriate reasoning and selection of the best answer behind the scenes, and replies with only one sentence (or even phrase) which includes the answer to the question. Consequently, the ultimate QA system should be able to provide answers in a similar way.

7.1 Rephrasing the Collected Questions

The focus of this project is not so much the information available from the database, after all, negation is not domain specific, but rather the analysis of those structures which include negation, and the correct IR processes that will give the expected answer based on the structure of the input.

For this reason, the collected questions by email were revisited, aiming to see for how many questions the system would correctly analyse the input, conduct the negation algorithm, and provide the correct answer, if the relevant information was stored in the database.

There were 50 questions, from which 5 were completely out of the scope of the domain. Those 5 questions looked more like open-domain questions, which the system would not be able to capture, even with an extension of the database. These questions were thus removed from all the calculations, but have remained in the Appendix A list. These are the excluded questions:

- *Is there a famous director in Japan who was not born there?* (it is hard to find information about the birthplaces of directors)
- *Did Michael Moore produce any films with no critical mood?* (critical mood could be considered as a subjective evaluation, thus it is not classified)
- *Is Kathreen Zeta Jones' mother tongue not English?* (it is almost impossible to classify the mother tongues of people)
- *Are there any martial-arts movies where the characters are not Asian?* (even if there is information about the nationality of actors, still this will not include the continent which that country belongs to)
- *Is it not true that actors are very rich?* (rhetorical question; no further comment)

The revisiting of the questions has resulted in three classifications, the percent-

ages of which are shown in table 7.1.

1. The questions which NotFilms is able to analyse and conduct the negation algorithm on, without any rephrasing of the question (represented by the 'Exact Question' column).
2. The questions which NotFilms is able to capture and conduct the negation algorithm on, even with some rephrasing (added to the 'Exact Question' numbers and shown in column '+ Rephrasing').
3. The questions which NotFilms would be able to conduct the negation algorithm on, by allowing rephrasing, and if there was more information available in the database. This refers to questions which the system has the linguistic and retrieval processes to analyse, but lacks the relevant information from the database (added to the '+ Rephrasing' numbers and shown in column '+ Knowledge Extension').

	Exact Question	+ Rephrasing	+ Knowledge Extension
counts (total 45)	2	19	34
percentage %	4.44%	42.22%	75.55%

*Table 7.1: The amount of the Collected Sample Questions
for which NotFilms would give an answer*

Looking at the numbers in Table 7.1 there are some observations important for evaluation:

A. Small Database

It is obvious that the existing NotFilms system is very constrained by the present form of its database. Only for 42.22% of the total of questions the system would be able to produce the answer, even with some rephrasing. Extend-

ing the database is expected to increase the answerability rate of the system, as the third column of the table suggests: that with exactly the same linguistic and IR processes, and only by making more information available to the user, the answerability rate can increase drastically.

Nevertheless, as it was explained earlier, this project was not aiming to the development of a complete knowledge base. Besides, database development is on its own a whole different area. The purpose of NotFilms was rather to bring some lights to the area of negation, and how negative constructions can be dealt with by QA systems. Thus, the small size of the database is not something of an immediate threat to the contribution of the NotFilms system.

B. Negative Structures unmanageable by NotFilms

As can be recalled from the chapter on Negation Theory, there are many ways to express negation, with different syntactic constructions, morphological constructions, or with the use of specific words with negative meanings, yet which appear in declarative constructions. In the given set of collected questions, two of the above constructions can be depicted: Morphological Negation (non-English movie Vs English movie), as well as Syntactic Negation, which includes all the different constructions with the use of the adverb 'not' and 'no' (as in *...produced no documentaries*).

The same answer can be given to the same question, even if the structure of the sentence is not the same. This is what the rephrasing of the questions has shown: that by changing the structure of a question, writing it with other words, its meaning remains unchanged, and the rephrased question still requires the same answer.

The results show that from 4.5% of a sentence structure analysis, the percentage of answerability rises to 75.5%, just because the negative construction has been rephrased. This means that with the system's current negation processes, and if the answer existed in the database, the system is capable of answering 3/4 of the users' questions.

7.2 Users' Evaluation

In order to measure the validity and relativity of the answers which NotFilms would give for the rephrased questions, 5 native speakers of English were presented with the list of the actual questions paired with their rephrased counterparts, and the format of the answers which the system would give to the rephrased ones, and were asked to judge the following:

- A. the Relevance of the original question to the rephrased version
- B. the Relatedness of each answer (i.e. how related the answer of the rephrased question is to the answer which the users would expect for the original question)

The judgement was in the form of a questionnaire, in which the users were asked to make a choice from 1 (irrelevant rephrasing) to 5 (successful rephrasing) and another from 1 (irrelevant answer to the original question) to 5 (complete answer for the original question) for each case. The complete catalogue of what was shown to the users can be found in Appendix B.

Due to the small number of participants, no formal statistical tests were conducted. However, the range of answers is indicative to the following observations.

The responses of the users showed that out of the 32 sets of rephrased questions, 14 (43.75%) were rephrased successfully (referring to choices 4 and 5 of the Likert scale). As an extend, the same percentage of answers were judged as satisfactory for both the original and the rephrased version of the sentence (again referring to choices 4 and 5 of the Likert scale). These examples are the first 14 sets in the appendix. On the other hand, although the rephrases of the other 18 sentences (56.25%) were not judged as successful (choices 1 and 2), still, the responses of the system to the same 18 sentences were not judged as completely irrelevant to the original questions (choices 2, 3 and 4). These

findings reinforce the validity of NotFilms' answers: either with or without rephrasing of the questions, at least some answer, somewhat relevant, if not completely relevant to the original question, can be retrieved.

7.3 Conclusion

The NotFilms system with its linguistic and IR processes is able to capture the three thirds of users' questions, so long the users are patient enough to have more than one tries in rephrasing their original questions. On the other hand, NotFilms allows only a specific and narrow type of negative constructions, which forces the user to rephrase the original sentence in most of the times.

The evaluation of the current system shed light to various aspects of Negation. The rephrasing of the sentences has shown that a negative meaning can be maintained even with a change in the structure of the sentence. Rephrasing a sentence might cause some alteration of the meaning of the original sentence. However, an answer which completely satisfies the rephrased question can at least partly -if not fully- satisfy the original question as well.

Chapter 8

Conclusions

8.1 NotFilms Capabilities

The NotFilms system in its present form is able to answer positive and negative close domain questions with the same efficiency. Negation in the system is realised with the use of the 'not' particle before auxiliaries. The system parses the input and produces its semantic representation, using modules of an existing system called CURT. It then uses the negation algorithm and the IR processes to contact the knowledge base and retrieve the relevant answer based on the logical representation of the input.

The acceptable questions are subject or object questions, and are introduced with a wh-word (who, which, what). In case the input has been parsed, but no answer can be found, the user is notified that no relevant entries are available. If the system cannot parse an input, the user is asked to rephrase. Otherwise, if an answer is found, then the system presents the exact answer to the user within a list.

8.2 Next Steps - Future Work

There are two areas on which more work needs to be done. One relates to the kinds of information available in the knowledge base, and the other relates to the linguistic processes of negation.

Since the innovation of the system is the ability to treat negation, further additions to what is considered negation are needed. To begin with, the system only deals with *not*, yet still, not with all structures which can include that particular word. In order for the present system to have a start and an end, many syntactic constructions have been excluded from the linguistic analysis: yes/no questions, passive voice constructions, relative clauses, constructions or prepositional phrase constructions.

The linguistic modules will need to be extended to accept Yes/No questions. Even if a single yes or no is not very informative for an answer, still, this option should exist in the system. Passive Voice constructions and PP constructions are also related, stemming from the treatment of questions with prepositional phrases. However, additional semantic knowledge should be put in the lexicon, grammar, and semantic modules of the system, to realise which word is the agent of a sentence (the head of passive voice PPs e.g. by USA, by Spielberg, etc.), irrespectively from the syntactic position in the sentence. The acceptance of questions denoting place or time (introduced by *where*, *when*, etc. would also need additions to the present linguistic modules. These are very important additions, if we try to consider the disambiguation of the questions below:

1. *Which dramas were produced in 1995?*
2. *Which dramas were produced in Pakistan?*
3. *Which dramas were produced by Pakistan?*

None of these three questions is currently captured by NotFilms: all three of

them have a passive structure and include a prepositional phrase. An improved system will be able to understand that sentence 1 needs the dramas which were produced in a particular year, that sentence 2 needs dramas which were filmed in Pakistan, and that sentence 3 needs dramas which were produced by any Pakistani production company. Hence, an extended linguistic component would provide the IR component with more semantic readings, so that the appropriate negation processes could be applied to those different questions as well.

Looking at the original questions which have been rephrased, it is obvious that all of them needed rephrasing due to the lack of the linguistic component to produce the semantic reading of a certain construction. By extending the linguistic coverage of the system, more sentence constructions will be accepted, and the users will not be required to rephrase almost all of their sentences. Consequently, the time required for each question to be answered will be drastically reduced.

Regarding the knowledge of the system, the present database includes very little information. It is a requirement to have more information available, so to be able to test the linguistic and IR processes on them. Having more kinds of information in the knowledge base, like for example different years of production, nationalities of actors and actresses, the country in which each production company is located, information about costume designers, soundtracks, etc. offers more choices on different questions. The more things people have to talk about, the more constructions people are able to use.

8.3 Conclusion

The present project has been an effort to bring Negation to the attention of current Question Answering research. At the beginning, there was an introduction to the theory of negation and its linguistic realisations. Even in theory,

negation has not been dealt with as an independent and complete subject on its own. It has always been seen in comparison to affirmation, and has not been analysed to the same extend as the positive counterpart.

There was also an introduction to IR systems, and question answering systems. The lack of negation treatments was pointed out, in order to continue with a suggestion as to how negative input can be treated. For the purposes of the present project, a new QA system was developed, called NotFilms. It is a restricted-domain QA system, which answers subject and object questions about movies. It accepts positive questions, as well as negative ones, which denote negation with the particle *not* attached to the auxiliaries. NotFilms reads the natural language input, produces the logical form of the question, uses the IR processes based on the logical representation of the input in order to retrieve the answer from the knowledge base, and makes the answer(s) available to the user.

More and continuous work needs to be done in the area of Negation and its inclusion in current Question Answering systems. Although affirmative questions are indeed more common, for the moment, negation is treated as a nonexistent phenomenon in natural language. NotFilms has been an attempt to introduce Negation into the area of Question Answering. I hope that this project will trigger the interests of scholars, researchers and students, and prompt them to develop techniques, which will capture negation in broader sense in Question Answering.

Appendix A

List of Questions

The questions which served as the starting point for the development of the NotFilms system:

- *Questions in Passive Voice*
 1. Which English movies were not filmed in English-speaking countries?
 2. How many movies were casted in Spain and were not in Spanish?
 3. Which Spanish films were not translated in English?
 4. Are there any countries in which neither films nor documentaries were produced?
 5. Were there no western-like movies produced in Bolywood?
 6. Which movies were ranked among the top 200 and were not made by American production companies?
 7. Were there no documentaries produced by Warner Bros?
 8. Are there any Warner Bros films that were not produced in USA?
 9. Which Japanese thrillers were not produced in the 1980s?
 10. How many thrillers were not produced after 1995?
 11. Which of the films were not filmed in the United States?

12. Who wasn't the lead actress on a film directed by Spielberg?
 13. Was there no actor from subcontinent in the films that were not filmed in the USA?
 14. Is there a famous director in Japan who was not born there?
- *Questions in Active Voice*
 1. Which movies took place in the US and were not in English?
 2. Which production companies did not produce any English films?
 3. Were any films produced in countries that did not speak English?
 4. Which English thrillers did Spielberg not direct?
 5. Who directed non-English movies?
 6. Did Spielberg direct any non-English films?
 7. Did Akira Kurosawa use a cast which did not consist of Japanese actors?
 8. Did Michael Moore direct a film which was not a documentary?
 9. Did Michael Moore produce any films with no critical mood?
 10. Which Greek black-and-white movies did Aliki Vougiouklaki not play in?
 11. Were there any actors who played in T.V. movies but did not play in t.v. series?
 12. Did Jerry Lewis star in films which were not comedies?
 13. In which non-English movies did Salma Hayek play?
 14. Which actors already have a career in Bollywood but are not known in Hollywood yet?
 15. Has Kylie Minogue participated in no other soap operas besides Neighbours?
 16. Which years after 1980 did Walt Disney not produce any films?
 17. Which country had dramas but no comedies?
 18. How many countries produced no documentaries?
 19. Which companies did not produce any dramas?
 20. Are there any countries that did not produce any comedies?

21. Who wasn't the lead actress on a film directed by Spielberg?
 22. Which actresses have won a second best actor Oscar award, but not a first one?
 23. Which country has no progress in movie production so far?
- *Questions with the verb 'to be' ('is', 'are', 'was', etc.)*
 1. Which movies took place in the US and were not in English?
 2. How many movies were casted in Spain and were not in Spanish?
 3. How many films are not in English?
 4. Which Hollywood movies are not in English?
 5. Were there any James Bond films that did not include the word 'die'?
 6. Of which American horror movies the director was not Spielberg?
 7. Which Spielberg movies were not horror?
 8. Are there any martial-arts movies where the characters are not Asian?
 9. Are there any countries in which neither films nor documentaries were produced?
 10. Were there no western-like movies produced in Bollywood?
 11. Which top 200 movies were not in English?
 12. Did Michael Moore direct a film which was not a documentary?
 13. Which of Spielberg movies were not romantic-dramas?
 14. Were there no documentaries produced by Warner Bros?
 15. Are there any Warner Bros films that were not produced in USA?
 16. Were there any actors who played in T.V. movies but did not play in t.v. series?
 17. Is Leonardo Di Caprio not of American origin?
 18. Is Kathreen Zeta Jones' mother tongue not English?
 19. Is it not true that actors are very rich?
 20. Is there a famous director in Japan who was not born there?
 21. Did Jerry Lewis star in films which were not comedies?

22. Are there any countries that did not produce any comedies?
23. Was there no actor from subcontinent in the films that were not filmed in the USA?
24. Was there no movie production in Greece before the 1950s?

Appendix B

The entries which were presented to evaluators

Each question in bold is the original sample question, each question in italics is the rephrasing of the first one (the one which is posed to the system), and what follows is the kind of answer the system would give to the rephrased question.

1. **Which English movies were not filmed in English-speaking countries?**

Which country filmed English films?

The answer is a list of all the countries in which English films were filmed.

2. **Are there any countries in which neither films nor documentaries were produced?**

Which countries didn't produce documentaries and didn't produce films?

No entries are shown because 'documentaries' are classified as a genre of films.

3. **Were there no western-like movies produced in Bollywood?**

Which western films did India produce?

The answer is a list of the names and years of production of western movies produced in India.

4. Were there no documentaries produced by Warner Bros?

Which documentaries did Warner Bros produce?

The answer is a list of documentaries produced by Warner Bros.

5. Are there any Warner Bros films that were not produced in USA?

Which Warner Bros films didn't USA film?

The answer is a list of Warner Bros films, which were filmed outside the United States.

6. Which of the films were not filmed in the United States?

Which films didn't USA film?

The answer includes a list of films which were not filmed in USA, along with their Year of production.

7. Who wasn't the lead actress on a film directed by Spielberg?

Who didn't star in a Steven Spielberg film?

The answer includes all the actors who did not have a starring role in a Spielberg movie.

8. Were there any actors who played in T.V. movies but did not play in T.V. series?

Which actors played in TV movies and did not play in TV series?

The answer includes the actors who played in T.V. movies and did not play in T.V. series.

9. Has Kylie Minogue participated in no other soap operas besides Neighbours?

Which soap opera did Kylie Minogue participate in?

The answer includes all the soap operas which Kylie Minogue participated in.

10. Which country had dramas but no comedies?

Which country produced dramas and did not produce comedies?

The answer is a list of countries where dramas and not comedies were produced.

11. How many countries produced no documentaries?

Which countries didn't produce any documentaries?

The answer includes the countries which produced no documentaries.

12. Are there any countries that did not produce any comedies?

which countries did not produce any comedies?

The answer is a list of countries which did not produce comedies.

13. Which actresses have won a second best actor Oscar award, but not a first one?

which actresses won a second best actor oscar award and didn't win a first best actor oscar award?

The answer includes the names of actresses who won a second best actor oscar award and didn't win a first best actor oscar award.

14. Are there any countries in which neither films nor documentaries were produced?

Which country didn't produce films?

The answer includes the countries which did not produce films and didn't produce documentaries (the system realises documentaries as a kind of a film)

— — — — — — —

15. How many movies were casted in Spain and were not in Spanish?

Which English movies did Spain film?

The answer gives movies in English, which were filmed in Spain.

16. Which movies took place in the US and were not in English?

Which German or Greek movies did USA film?

The answer gives the complete list of German and Greek movies filmed in the United States.

17. Were any films produced in countries that did not speak English

Which films didn't England or USA or Ireland or New Zealand or Australia film?

The answer gives the films which were not filmed in any of the above countries.

18. Who directed non-English movies?

Who did not direct English movies?

The answer includes the names of people who did not direct any English movie.

19. Did Spielberg direct any non-English films?

Which German movies and which Greek movies did Steven Spielberg direct?

The answer gives the German and Greek movies which Spielberg direct.

20. Did Akira Kurosawa use a cast which did not consist of Japanese actors?

Which actors participated in an Akira Kurosawa movie?

The answer includes all the actors who played in a Kurosawa movie.

21. Did Michael Moore direct a film which was not a documentary?

Which dramas or thrillers or science fiction movies or action films did Michael Moore produce?

The answer includes the films of the above genres directed by Moore.

22. Which Greek black-and-white movies did Aliki Vougiouklaki not play in?

Which greek movies didn't aliki vougiouklaki play in?

The answer includes all the names and years of movies in which Vougiouklaki did not play.

23. Did Jerry Lewis star in films which were not comedies?

Which dramas or thrillers or musicals did Jerry Lewis star in?

The answer includes names and years of films of the above genres in which Lewis starred.

24. In which non-English movies did Salma Hayek play?

Which German or English movies did Salma Hayek play in?

The answer shows German and English movies in which Hayek played.

25. Which actors already have a career in Bolywood but are not known in Holywood yet?

Who starred in Indian movies and did not play in any english movies?

The answer gives names of actors who starred in Hindi movies and did not appear in any English movie.

26. Which years after 1980 did Walt Disney not produce any films?

What did Walt Disney produce?

The answer consists of titles and years of movies produced by Disney.

27. Which country has no progress in movie production so far?

Which country filmed films and did not produce films?

The answer shows countries in which movies were filmed, but which themselves did not produce any films.

28. How many films are not in English?

Which German and Greek movies didn't steven spielberg produce?

The answer includes German and Greek movies produced by Spielberg.

29. Which Hollywood movies are not in English?

Which German movies or Greek movies or Italian movies did USA produce?

The answer includes Greek and German and Italian movies produced by USA.

30. Of which American horror movies the director was not Spielberg?

Which English horror movies didn't spielberg direct?

The answer shows English speaking horror movies which Spielberg not direct.

31. Which Spielberg movies were not horror?

Which thrillers or dramas or science fiction movies or crime movies or musicals did Steven Spielberg direct?

The answer includes titles and years of films of the above genres, directed by Spielberg.

32. Which of Spielberg movies were not romantic-dramas?

Which movies and not dramas did Steven Spielberg direct?

The answer includes all the movies of Spielberg which were not dramas.

Bibliography

- Blackburn, P. and Bos, J. (2005). *Representation and Inference for Natural Language. A First Course in Computational Semantics*. Studies in Computational Linguistics Series. CSLI.
- Cardie, C., Ng, V., Pierce, D., and Buckley, C. (2000). Examining the role of statistical and linguistic knowledge sources in a general-knowledge question-answering. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, pages 180–187.
- Clark, H. H. (1976). *Semantics and Comprehension*. Mouton de Gruyter, Amsterdam.
- Clark, H. H. and Clark, E. (1977). *Psychology and Language: An introduction to Psycholinguistics*. Harcourt Brace Jovanovich, New York.
- den. Dikken, M. (1995). *Particles: On the Syntax of Verb-particle, Triadic, and Causative Constructions*. Oxford University Press, New York.
- Diekema, A. R., Yilmazel, O., and Liddy, E. D. (2004). Evaluation of restricted domain question-answering systems. In *ACL 2004 Annual Conference*, University of Pennsylvania. Association of Computational Linguistics.
- Downing, L. H. (2000). *Negation, Text Worlds, and Discourse: The Pragmatics of Fiction*, volume LXVI of *Advances in Discourse Processes*. Ablex, Stanford, Connecticut.
- Fukumoto, J., Kato, T., and Masui, F. (2003). Question answering challenge

- (qac-1): An evaluation of question answering tasks at ntcir workshop 3. In *Proceedings of the AAAI Spring Symposium: New Directions in Question Answering*, pages p.122–133.
- Givon, T. (1993). *English Grammar: A Function-Based Approach*. John Benjamins, Amsterdam.
- Gueron, J. (1987). Clause union and the verb-particle construction in english. In *North Eastern Linguistics Society Annual Meeting (NELS)*, number 17.
- Jordan, M. P. (1998). The power of negation in english: Text, context and relevance. *Journal of Pragmatics*, 29:705–752.
- Jurafsky, D. and Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, Upper Saddle River, New Jersey 07458.
- Klima, E. S. (1964). *The Structure of Language: Readings in the Philosophy of Language*, chapter Negation in English, pages 246–323. Prentice Hall, Hemel Hempstead.
- Maybury, M. T. (2002). Toward a question answering roadmap. Technical report, The MITRE Corporation, Bedford, MA 01730.
- Nyberg, E. and Mitamura, T. (2002). Evaluating qa systems on multiple dimensions. In *Proceedings of LREC 2002 Workshop on QA Strategy and Resources*, Las Palmas, Gran Canaria.
- Oesterle, J., editor (1962). *Aristotle: On Interpretation: Commentary by St. Thomas and Cajetan. (Sententia super Peri hermenias)*. Marquette University Press, Milwaukee.
- Pacitti, D. (1991). *The Nature of Negative: Towards an Understanding of Negation and Negativity*. Giardini, Pisa.
- Ramchand, G. and Svenovius, P. (2002). The lexical syntax and lexical semantics of the verb-particle construction. In Mikkelsen, L. and Potts, C., editors, *WCCFL 21 Proceedings*, pages 387–400, Somerville, MA. Cascadilla Press.

- Reyes, M. P., Macnamara, J., Reyes, G. E., and Zolfaghari, H. (1999). *What is Negation?*, chapter Models for Non-Boolean Negations in Natural Languages based on Aspect Analysis, pages 241–260. Kluwer Academic Publishers, Netherlands.
- Russell, S. and Norvig, P., editors (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall - Pearson Education International, Upper Saddle River, New Jersey 07458, second edition.
- Tappenden, J. (1999). *What is Negation?*, chapter Negation, Denial and Language Change in Philosophical Logic, pages 261–298. Kluwer Academic Publishers, Netherlands.
- Tottie, G. (1991). *Negation in English Speech and Writing: A Study in Variation*. Academic Press, San Diego.
- Volterra, V. and Antinucci, F. (1979). *Developmental Pragmatics*, chapter Negation in Child Language: A Pragmatic Study, pages 281–292. Academic Press, London.
- Wason, P. C. (1965). The contexts of plausible denial. *Journal of Verbal Learning and Verbal Behavior*, 4:7–11.
- Wurmbrand, S. (2000). The structure(s) of particle verbs.